FrameWork 2
Solución de Integración con Documentum

Versión 1.2

Área de Integración y Arquitectura de Aplicaciones

Agencia de Informática y Comunicaciones de la Comunidad de Madrid

VICEPRESIDENCIA Y PORTAVOCÍA DEL GOBIERNO
Comunidad de Madrid

Hoja de Control

Título	Solución de Integración con Documentum		
Documento de Referencia			
Responsable	Área de Integración y Arquitectura de Aplicaciones		
Versión	1.2	**Fecha Versión**	04/02/2013

Registro de Cambios

Versión	Causa del Cambio	Responsable del Cambio	Fecha del Cambio
1.0	Versión inicial del documento	Área de Integración y Arquitectura de Aplicaciones	06/10/2009
1.1	Se modifica el apartado 5.1.1 LIBRERÍAS NECESARIAS para añadir 3 librerías de las que depende docu_lib(certjFIPS.jar,jsafeFIPS.jar aspectjrt.jar)	Área de Integración y Arquitectura de Aplicaciones	19/04/2010
1.2	Adaptar documento a la plantilla estándar de los manuales de arquitectura. Modificación apartado 3 PUNTOS A CONSENSUAR ANTES DEL INICIO DEL PROYECTO Inclusión de todo lo refererente al nuevo servicio web de VerRendition	Área de Integración y Arquitectura de Aplicaciones	30/06/2011

Índice

MODIFICACIÓN APARTADO 3 PUNTOS A CONSENSUAR ANTES DEL INICIO DEL PROYECTO 2

1 INTRODUCCIÓN 5

1.1 AUDIENCIA OBJETIVO 5
1.2 CONOCIMIENTOS PREVIOS 5

2 DESCRIPCIÓN 5

2.1 SERVICIOS WEB DE ACCESO A DOCUMENTUM: DOCU_WS 6
2.2 LIBRERÍA DE ACCESO A DOCUMENTUM: DOCU_LIB 9
2.3 LIBRERÍA PROPIETARIA DE DOCUMENTUM: DFC's 10

3 PUNTOS A CONSENSUAR ANTES DEL INICIO DEL PROYECTO 10

4 SOLUCION DE INTEGRACION DOCU_WS 12

4.1 INSTALACIÓN Y CONFIGURACIÓN 12
4.1.1 LIBRERIAS NECESARIAS 12
4.1.2 INSTALACION 12
4.1.3 CONFIGURACION 13
4.2 USO 15
4.2.1 PROGRAMACIÓN DE CLIENTES 15
4.2.2 GENERACIÓN DE FICHEROS XML DE ENTRADA A LOS SERVICIOS WEB. 15
4.2.3 URL'S DE ACCESO 15
4.2.4 USO DEL SERVICIO WEB DE ADMINISTRACION DE GRUPOS 15
4.2.5 USO DEL SERVICIO WEB DE BORRAR DOCUMENTO 19
4.2.6 USO DEL SERVICIO WEB DE PEDIR RENDITION 22
4.2.7 USO DEL SERVICIO WEB DE VER DOCUMENTO 25
4.2.8 USO DEL SERVICIO WEB DE TABLAS EXTERNAS 27
4.2.9 USO DEL SERVICIO WEB DE CHECK DOCUMENTO 32
4.2.10 USO DEL SERVICIO WEB DE IMPORTAR DOCUMENTO 39
4.2.11 USO DEL SERVICIO WEB DE BUSCAR DOCUMENTOS 43
4.2.12 USO DEL SERVICIO WEB DE MODIFICAR PERMISOS DE UN DOCUMENTO 46
4.2.13 USO DEL SERVICIO WEB DE GESTION DE CARPETAS 49
4.2.14 USO DEL SERVICIO WEB DE VER DOCUMENTO 52
4.3 PAQUETES COMUNES 55
4.3.1 Paquete docu_ws.ws.icm.config 55
4.3.2 Paquete docu_ws.ws.icm.util.constants 55
4.3.3 Paquete docu_ws.ws.icm.util.exception 56
4.3.4 Paquete docu_ws.ws.icm.util.WebServices 56
4.3.5 Paquete docu_ws.ws.icm.util.logs 56
4.3.6 Paquete docu_ws.ws.icm.xml 56

5 SOLUCION DE INTEGRACION DOCU_LIB 57

5.1 INSTALACION Y CONFIGURACION 57
5.1.1 LIBRERÍAS NECESARIAS 57
5.1.2 INSTALACION 57
5.1.3 CONFIGURACION 57
5.2 USO 58
5.2.1 GENERAL 58
5.2.2 GENERACIÓN DE FICHEROS XML DE ENTRADA A LOS SERVICIOS 58
5.1.1 USO DEL SERVICIO DE ADMINISTRACION DE GRUPOS 59
5.1.2 USO DEL SERVICIO DE BORRAR DOCUMENTO 63
5.1.3 USO DEL SERVICIO DE PEDIR RENDITION 66

5.1.4 USO DEL SERVICIO DE VER DOCUMENTO ... 69
5.1.5 USO DEL SERVICIO DE TABLAS EXTERNAS ... 72
5.1.6 USO DEL SERVICIO DE CHECK DOCUMENTO ... 76
5.1.7 USO DEL SERVICIO DE IMPORTAR DOCUMENTO ... 84
5.1.8 USO DEL SERVICIO DE GESTION DE CARPETAS ... 88
5.1.8.1 DESCRIPCION DEL METODO ... 88
5.1.9 USO DEL SERVICIO DE VER RENDITION ... 92
5.2 PAQUETES COMUNES ... 94
5.2.1 PAQUETE DOCU_LIB.ICM.CONFIG ... 94
5.2.2 PAQUETE DOCU_LIB.ICM.UTIL.CONSTANTS ... 95
5.2.3 PAQUETE DOCU_LIB.ICM.UTIL.EXCEPTION ... 95
5.2.4 PAQUETE DOCU_LIB.ICM.XML ... 95
6 SOLUCION DE INTEGRACION DIRECTA CON DFC'S ... 96
6.1 INSTALACION Y CONFIGURACION ... 96
6.1.1 LIBRERÍAS NECESARIAS ... 96
6.1.2 INSTALACION ... 96
6.1.3 CONFIGURACIÓN ... 96
6.1.4 USO ... 97
6.1.5 RECOMENDACIONES Y BUENAS PRÁCTICAS ... **¡ERROR! MARCADOR NO DEFINIDO.**
7 LIBRERÍA DOCU_UTIL_LIB. TRATAMIENTO DE FICHEROS XML ... 98
7.1 GENERACIÓN XML DE IDA ... 98
7.2 MANEJO DEL XML DE VUELTA ... 99
8 PREGUNTAS MÁS FRECUENTES ... 99
9 ENLACES RELACIONADOS ... 100

Índice

1 INTRODUCCIÓN

Este documento contiene el manual de utilización de los Servicios de Gestión Documental proporcionados por Icm para el Framework 2.

A lo largo del documento existem hipervínculos a otros documentos relacionados, publicados en SOJA. La url utilizada es la url de Internet (https://gestiona.madrid.org/soja_int) para la cual se requiere usuario y password, para aquellos usarios que accedan a SOJA vía Intranet, esta url debe sustituirse por (http://desarrollo.madrid.org/soja_int)

1.1 AUDIENCIA OBJETIVO

El lector objetivo de este documento es toda aquella persona que esté desarrollando una aplicación web basada en el framework 2 y necesite acceder al gestor documental de Documentum.

1.2 CONOCIMIENTOS PREVIOS

Para un completo entendimiento del documento, el lector deberá tener conocimientos previos sobre las siguientes tecnologías:

- Lenguaje Java
- Api DFC de Documentum
- Tipos Documentales
- DQL
- Metodología de Desarrollo de Documentum
- Webservices

Para saber más sobre dichas tecnologías, consultar el apartado de este documento, *Enlaces Relacionados*.

2 DESCRIPCIÓN

Desde las aplicaciones que se desarrollen con el framework 2 se puede acceder al gestor documental utilizando alguna de las siguientes soluciones:

- Servicios web de acceso a Documentum: **DOCU_WS**
- Librería de acceso a Documentum: **DOCU_LIB** (requiere autorización).
- Librería propietaria de Documentum: **DFC's** (requiere autorización).

Independientemente de la solución elegida se desarrollará el Modelo de Datos Documental según el documento de Normativa_de_desarrollo_ICM_con_Documentum publicado en https://gestiona.madrid.org/soja_int/html/web/EnlaceManual.icm?cd_elemento=1344

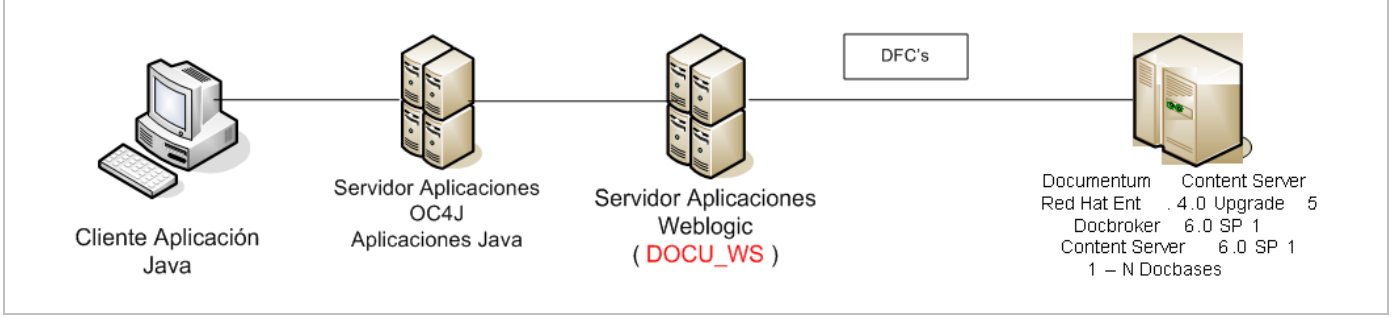
2.1 SERVICIOS WEB DE ACCESO A DOCUMENTUM: DOCU_WS

Icm proporciona un conjunto de operaciones básicas con documentum a través de los ***Servicios web de Icm*** (docu_ws).

Como norma general se considerará obligatorio el uso de docu_ws como solución de acceso a documentum desde aplicaciones J2EE.

En el siguiente gráfico se muestra la integración del aplicativo J2EE con Documentum mediante los servicios web (docu_ws).

ACCESO A DOCUMENTUM MEDIANTE SERVICIOS WEB


DFC's
Cliente Aplicación Java
Servidor Aplicaciones OC4J Aplicaciones Java
Servidor Aplicaciones Weblogic (DOCU_WS)
Documentum Content Server
Red Hat Ent . 4.0 Upgrade 5
Docbroker 6.0 SP 1
Content Server 6.0 SP 1
1 – N Docbases

Cada uno de los Servicios Web, previamente a realizar la operación que le corresponda, realiza la conexión con el repositorio, y posteriormente se desconecta del mismo.

Los Servicios Web que componen esta solución son los siguientes:

- ***VerDocumento***. Este WebService realiza la visualización del documento que se corresponde con el identificador del documento que se le pasa como parámetro en un xml. El retorno de este WebService es un objeto en el cual en su primera posición está el Datahandler del contenido del documento, que el cliente se encargará de transformar en un fichero físico para guardar en local y visualizarlo, y en la segunda posición devuelve un xml en el que nos indica el nombre del documento indicado.

    - ***Operación***: ***verDocumento.*** La forma de llamar a este Web Service es llamar al servicio VerDocumento en su operación **verDocumento**, a la cual se le pasará directamente el identificador del documento

- ***BorrarDocumento.*** Este WebService realiza el borrado del documento que se corresponde con el identificador del documento que se le pasa como parámetro en un xml.

    - ***Operación: borrarDocumento.*** La forma de llamar a este Web Service es llamar al servicio de BorrarDocumento en su operación **borrarDocumento**, a la cual se le pasará directamente el

identificador del documento

- ***BuscarDocumento.*** Este WebService permite realizar consultas en documentum.

    - ***Operación: buscarDocumento.*** La forma de llamar a este Web Service es llamar al servicio BuscarDocumento en su operación **buscarDocumento**, a la cual se le pasará una consulta dql como parámetro en un xml

- ***GestionTablas.*** Este WebService permite realizar modificaciones, inserciones y borrados en tablas externas.

    - ***Operación: gestionTablas.*** La forma de llamar a este Web Service es llamar al servicio GestionTablas en su operación **gestionTablas**, a la cual se le pasará la operación a realizar como parámetro en un xml

- ***PedirRendition.*** Este WebService permite generar transformaciones de documentos a formatos .pdf y .html.

    - ***Operación: pedirRendition. .*** La forma de llamar a este Web Service es llamar al servicio PedirRendition en su operación **pedirRendition**, a la cual se le pasará el identificador del documento a transformar y el formato deseado como parámetros en un xml

- ***AdministracionGrupos.*** Este WebService permite crear y modificar grupos

    - ***Operación: administrarGrupo.*** La forma de llamar a este Web Service es llamar al servicio AdministracionGrupos en su operación **administarGrupo**, a la cual se le pasara el identificador del grupo para el caso de modificar ó el nombre del grupo para el caso de crear uno nuevo, como parámetros en un xml

- ***ImportarModificar.*** Este WebService permite importar y modificar documentos. Permite modificar tanto metadatos como contenido.

    - ***Operación: ImportarModificarDoc.*** La forma de llamar a este Web Service es llamar al servicio ImportarModificar en su operación **ImportarModificarDoc**, a la cual se le pasara el identificador del documento (sólo en el caso de una modificación), tipo documental, lista de atributos, localización,… como parámetros en un xml y un DataHandler con el contenido del documento (sólo en el caso de una modificación).

- ***CheckDocumento*** Este WebService permite bloquear, desbloquear y registrar los cambios en un documento por un usuario previamente bloqueado por el mismo.

- ***Operaciones:***

  - ***Checkout:*** Esta operación nos permite bloquear un documento, para ello se le pasa el identificador del documento como parámetro en un xml. El retorno de este WebService es un objeto en el cual en su primera posición está el Datahandler del contenido del documento, que el cliente se encargará de transformar en un fichero físico para guardar en local, y en la segunda posición devuelve un xml en el que nos indica el nombre del documento bloqueado.

  - ***Checkin:*** Esta operación nos permite registrar los cambios que ha realizado un usuario en un documento previamente bloqueado por el mismo. Esta operación recibirá como parámetros de entrada un Datahandler con el contenido del documento y un xml donde se indicará obligatoriamente el identificador del documento, y donde se puede indicar opcionalmente la versión, etiqueta de versión, nueva extensión, descripción , opción para mantener ó no el bloqueo, opción para subscripción al documento y opción de indexado del documento.

  - ***cancelCheckout:*** Esta operación nos permite desbloquear un documento previamente bloqueado con la operación Checkout, para ello se le pasa el identificador del documento como parámetro en un xml.

- ***GestionCarpetas.*** Este WebService permite añadir, modificar y borrar cabinet y fólder.

  - ***Operación: gestionCarpetas.*** La forma de llamar a este Web Service es llamar al servicio GestionCarpetas en su operación **gestionCarpetas**, a la cual se le pasara el identificador del objeto (en el caso de modificación o borrado) o el nombre del objeto y tipo de objeto (en el caso de alta) como parámetro en un xml. Dentro de este xml también se le pueden pasar opcionalmente otros parámetros como path del padre, acl, indicador de forzado de borrado en caso de tener contenido y valores para los metadatos

- ***PermisosDocumento.*** Este Web Service permite modificar los distintos permisos de los usuarios y grupos asociados a un determinado documento, por lo tanto, lo que realmente se hace es cambiar la Acl asociada de dicho documento. Si cuando se va a modificar los permisos de un documento este contiene una acl estática se cambiará la acl del documento. Si el documento tiene una acl dinámica se cambiarán los permisos de la acl, manteniendo el documento esta acl.

  - ***permisosDocumento.*** La forma de llamar a este Web Service es llamar al servicio PermisosDocumento en su operación permisosDocumento, a la cual se le pasara el identificador del documento y la lista de permisos a asignar al documento como parámetros en un xml.

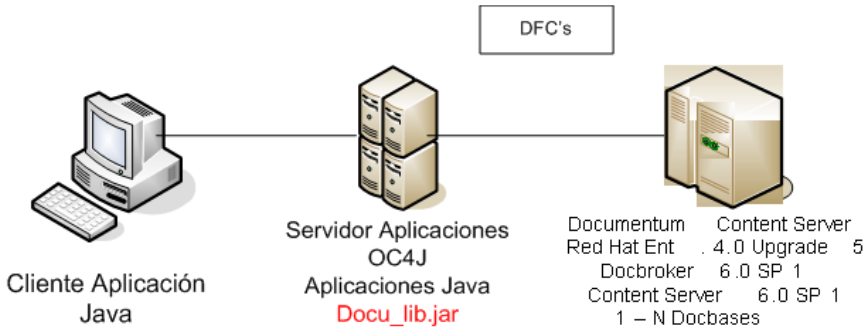
- ***VerRendition***. Este WebService realiza la visualización de la renderización que se corresponde con el identificador de renderización que se le pasa como parámetro en un xml. El retorno de este WebService es un objeto en el cual en su primera posición está el Datahandler del contenido de la renderización del documento, que el cliente se encargará de transformar en un fichero físico para guardar en local y visualizarlo, y en la segunda posición devuelve un xml en el que nos indica el nombre del documento indicado.

    - ***Operación***: ***verRendicion.*** La forma de llamar a este Web Service es llamar al servicio VerRendition en su operación **verRendicion**, a la cual se le pasará directamente el identificador de la renderización.

2.2 LIBRERÍA DE ACCESO A DOCUMENTUM: DOCU_LIB

Esta librería es una réplica de las funcionalidades de los Servicios Web
La llamada a cada uno los servicios se hace mediante la clase y método de cada uno de ellos, pasándoles como parámetro un fichero xml igual que en el caso de los servicios Web y un objeto Logger, mediante el cual se grabarán los posibles errores en un archivo .log.
Cada uno de los Servicios, previamente a realizar la operación que le corresponda, realiza la conexión con el repositorio, posteriormente se desconecta del mismo.
Para la correcta ejecución de los proyectos que utilicen esta solución, deben incorporar las DFC's de Documentum en el .ear de despliegue y un fichero de configuración dfc.properties dentro del directorio classes.
El siguiente gráfico muestra la integración del aplicativo J2EE con Documentum mediante la librería docu_lib

ACCESO A DOCUMENTUM MEDIANTE DOCU_LIB



2.3 LIBRERÍA PROPIETARIA DE DOCUMENTUM: DFC's

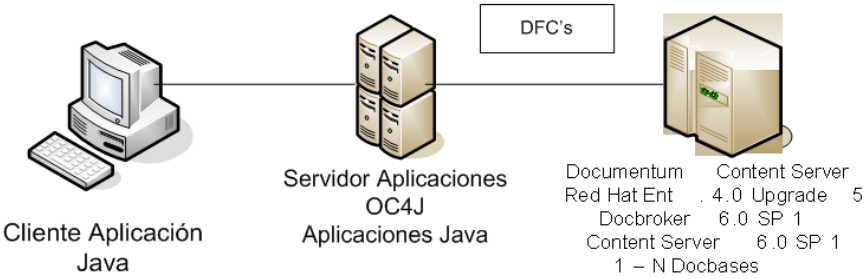
Cuando el Framework de servicios documentales ofrecido por ICM, no proporcione la funcionalidad requerida para el desarrollo del proyecto, podrán utilizarse las DFC'S proporcionadas por Documentum

Las DFC's también se utilizarán en el desarrollo de Java Médodos y en las Personalizaciones del Cliente Estándar de Documentum (Webtop).

Para la correcta ejecución de los proyectos que utilicen esta solución, deben incorporar las DFC's de Documentum en el .ear de despliegue y un fichero de configuración dfc.properties dentro del directorio classes

El siguiente gráfico muestra la integración del aplicativo J2EE con Documentum mediante las **(DFC's)** proporcionadas por el producto.

ACCESO A DOCUMENTUM MEDIANTE DFC's


DFC's
Cliente Aplicación Java
Servidor Aplicaciones OC4J Aplicaciones Java
Documentum Content Server
Red Hat Ent . 4.0 Upgrade 5
Docbroker 6.0 SP 1
Content Server 6.0 SP 1
1 – N Docbases

3 PUNTOS A CONSENSUAR ANTES DEL INICIO DEL PROYECTO

Previo al inicio del proyecto, existen una serie de puntos que deben de quedar perfectamente consensuados entre ICM y el proveedor. Dichos puntos son los siguientes:

- **Solución de Acceso a Documentum**. A excepción del uso de la solución de acceso a documentum vía WebService (DOCU_WS), el resto de soluciones deben ser consensuadas y autorizadas previamente por parte de Arquitectura de Aplicaciones.

- **Productos de Documentum a utilizar**: Se detallarán todos los productos de documentum que serán necesarios para la implementación de la aplicación. En la actualidad los productos adquiridos por icm son:
  - **Documentum Content Server**
  - **Documentum Index Server.**
  - **Documentum Webtop**
  - **Documentum Administrator**
  - **Documentum Document Transformation Services (DTS)**
  - **Documentum Business Process Manager**
  - **Documentum Web Publisher**
  - **Documentum Trusted Content Services**

- **Documentum Site Caching Services**
- **Documentum Retention Policy Services Language Packs**
- **EMC Documentum Reporting Services**

- **Volumetría de la aplicación**: Para dimensionar el espacio asignado en disco, deberá estimarse el tamaño medio por documento, número de documentos iniciales, y el incremento de los mismos por año.

- **Número estimado de usuarios finales de la aplicación**. Se debe de estimar el número de usuarios de la aplicación, además de la concurrencia esperada.

- **Autenticación y autorización de acceso a Documentum**: se debe indicar si el acceso va a ser con usuario personalizado o con usuarios genéricos. Además se indicará, información sobre los distintos tipos de roles que se vayan a crear y los permisos de cada uno de los roles.

- **Trazabilidad de datos de alto nivel de seguridad**: se debe indicar si los tipos documentales que se van a crear contienen datos personales de nivel alto, y si la trazabilidad se va realizar desde la aplicación ó desde documentum.

Cualquier otra identificación necesaria para el proyecto que no quede reflejada en el alcance de este documento, tiene que ser consensuada con el equipo de ICM antes del inicio del proyecto.

4 SOLUCION DE INTEGRACION DOCU_WS

4.1 INSTALACIÓN Y CONFIGURACIÓN

4.1.1 LIBRERIAS NECESARIAS

Las librerías necesarias para el correcto funcionamiento de las aplicaciones j2ee según Framework 2 que utilizan los Servicios Web propietarios de icm para acceder a Documentum son las descritas en la siguiente tabla:

Nombre	Motivo	Localización en SOJA	C	E
docu_util_lib.jar	Para generar los xml que se le pasan como parámetro a los distintos servicios	https://gestiona.madrid.org/soja_int/run/j/EnlaceLibreria.icm?cd_elemento=723	X	X
jaxb_icm_1_3.jar	Manejo de xml	https://gestiona.madrid.org/soja_int/run/j/EnlaceLibreria.icm?cd_elemento=181	x	x
seguridad_cm_1_1.jar	Plantilla del Framework 2	https://gestiona.madrid.org/soja_int/run/j/EnlaceLibreria.icm?cd_elemento=541		x
sistemas2_4.jar	Plantilla del Framework 2	https://gestiona.madrid.org/soja_int/run/j/EnlaceLibreria.icm?cd_elemento=501		x
servlet.jar	Plantilla del Framework 2	Todas estas librerías se han empaquetado en **lib_dep_docu_ws.zip** y se han publicado en https://gestiona.madrid.org/soja_int/run/j/EnlaceLibreria.icm?cd_elemento=742	x	
jaxrpc.jar	Paquete de librerías Axis			x
axis.jar	Paquete de librerías Axis		x	x
commons-discovery-0.2.jar	Paquete de librerías Axis			x
commons-logging-1.0.4.jar	Paquete de librerías Axis			x
saaj.jar	Paquete de librerías Axis			x
wsdl4j-1.6.1.jar	Paquete de librerías Axis			x
log4j-1.2.15.jar	Paquete de librerías Axis			x
activation.jar	Relacionada con el servicio de Importación de documentos (Datahandler)		x	x
apache_soap-2_3_1.jar	Relacionada con el servicio de Importación de documentos (Datahandler)			x
mail-1.4.jar	Relacionada con el servicio de Importación de documentos (Datahandler)		x	x

C: Librerías utilizadas en Compilación.
E: Librerías utilizadas en Ejecución.
Nota: Además de estas librerías se podrán utilizar cualquier librería autorizada para el Framework 2

4.1.2 INSTALACION

Para poder desarrollar aplicaciones j2ee con acceso a Documentum utilizando los Servicios Web proporcionados por Icm, es necesario que el proveedor instale estos Servicios Web en sus instalaciones.

Para ello se ha colgado en Soja (https://gestiona.madrid.org/soja_int) lo siguiente:

- ***Manual de instalación en el entorno del proveedor de los servicios Web***, publicado en: https://gestiona.madrid.org/soja_int/run/j/EnlaceManual.icm?cd_elemento=1341

- ***docu_ws.war.*** Fichero .war con los Servicios Web, publicado en: https://gestiona.madrid.org/soja_int/run/j/EnlaceRecurso.icm?cd_recurso=1082

4.1.3 CONFIGURACION

Las variables que pueden venir en el fichero de configuración de la aplicación son las indicadas en la siguiente tabla

Variables	Descripción	Valores que toma	Obligatorio
documentum.docbase	Nombre del repositorio	desdocum_01	SI
documentum.usuario	Normalmente la autenticación en documentum se realizará a través de LDAP. Esta variable sólo vendrá en el caso de necesitar un usuario genérico de conexión con Documentum.	xxxx_<usuario> siendo xxxx el nombre del proyecto	NO
documentum.clave	Clave del usurio encriptada del usuario genérico.		NO
documentum.docu_ws_url	Apunta a la url donde están desplegados los Web Services del Framework Documentum	http://icmaplic31:7004/docu_ws/services	NO
documentum.ws.adminService.url	Apunta a la url del Web Sevice Admin Service	%documentum.docu_ws_url%AdminService	NO
documentum.ws. adminService.metodo	Nombre del método del Web Service Admin Service	Admin Service	NO
documentum.ws.pedirRendition.url	Apunta a la url del Web Sevice Pedir Rendición	%documentum.docu_ws_url%PedirRendition	NO
documentum.ws. pedirRendition.metodo	Nombre del método del Web Service Pedir Rendición	pedirRendition	NO
documentum.ws.login.url	Apunta a la url del Web Sevice Login	%documentum.docu_ws_url%Login	NO
documentum.ws.login.metodo	Nombre del método del Web Service Login	getSessionUsuario	NO
documentum.ws.logout.url	Apunta a la url del Web Sevice Logout	%documentum.docu_ws_url%Logout	NO
documentum.ws.logout.metodo	Nombre del método del Web Service Logout	disconectSession	NO
documentum.ws.buscarDocumentos.url	Apunta a la url del Web Sevice Buscar Documentos	%documentum.docu_ws_url%BuscarDocumentos	NO
documentum.ws.buscarDocumentos.metodo	Nombre del método del Web Service BuscarDocumentos	buscarDocumentos	NO
documentum.ws.gestionCarpetas.url	Apunta a la url del Web Sevice Gestión Carpetas	%documentum.docu_ws_url%GestionCarpetas	NO
documentum.ws. gestionCarpetas.metodo	Nombre del método del Web Service Gestión Carpetas	gestionCarpetas	NO
documentum.ws.administracionGrupos.url	Apunta a la url del Web Sevice Administración Grupos	%documentum.docu_ws_url%AdministracionGrupos	NO

documentum.ws.administracionGrupos.metodo	Nombre del método del Web Service Administración Grupos	administrarGrupo	NO
documentum.ws.version.url	Apunta a la url del Web Sevice Versión	%documentum.docu_ws_url%Version	NO
documentum.ws.version.metodo	Nombre del método del Web Service Versión	getVersion	NO
documentum.ws.checkin.url	Apunta a la url del Web Sevice Check Documento	%documentum.docu_ws_url%CheckDocumento	NO
documentum.ws.checkin.metodo	Nombre del método del Web Service checkin	checkin	NO
documentum.ws.checkout.url	Apunta a la url del Web Sevice Check Documento	%documentum.docu_ws_url%CheckDocumento	NO
documentum.ws.checkout.metodo	Nombre del método del Web Service checkout	checkout	NO
documentum.ws.cancelCheckout.url	Apunta a la url del Web Sevice Check Documento	%documentum.docu_ws_url%CheckDocumento	NO
documentum.ws.cancelcheckout.metodo	Nombre del método del Web Service cancel checkout	cancelcheckout	NO
documentum.ws.gestionTablas.url	Apunta a la url del Web Sevice Gestión Tablas	%documentum.docu_ws_url%GestionTablas	NO
documentum.ws.gestionTablas.metodo	Nombre del método del Web Service Gestión Tablas	gestionTablas	NO
documentum.ws.verDocumento.url	Apunta a la url del Web Sevice Ver Documento	%documentum.docu_ws_url%VerDocumento	NO
documentum.ws.verDocumento.metodo	Nombre del método del Web Service Ver Documento	verDocumento	NO
documentum.ws.permisosDocumento.url	Apunta a la url del Web Sevice Permisos Documento	%documentum.docu_ws_url%PermisosDocumento	NO
documentum.ws.permisosDocumento.metodo	Nombre del método del Web Service Permisos Documento	permisosDocumento	NO
documentum.ws.borrarDocumento.url	Apunta a la url del Web Sevice Borrar Documento	%documentum.docu_ws_url%BorrarDocumento	NO
documentum.ws.borrarDocumento.metodo	Nombre del método del Web Borrar Documento	borrarDocumento	NO
documentum.ws.importarModificar.url	Apunta a la url del Web Sevice Importar Modificar	%documentum.docu_ws_url%ImportarModificar	NO
documentum.ws. importarModificar.metodo	Nombre del método del Web Sevice Importar Modificar	ImportarModificarDoc	NO

NOTA:
Todas las variables se declararán en el apartado
/*************************/
/* DOCUMENTUM */
/*************************/.

Cualquier otra variable que se necesite definir seguirá la siguiente nomenclatura **documentum.<descripción_variable>**

4.2 USO

4.2.1 PROGRAMACIÓN DE CLIENTES

Para la llamada a los WebServices, se deben programar una serie de clientes SOAP siguiendo la arquitectura propuesta por el Framework de ICM, en el documento **Creación WebServices** publicado en:
https://gestiona.madrid.org/soja_int/run/j/EnlaceManual.icm?cd_elemento=401
Dentro del paquete principal de la aplicación se creará un paquete llamado **clientes**, en el que se ubicarán los Clientes programados.
Esta serie de clientes deben invocar los servicios web pasándoles por parámetro el xml que contiene la información correspondiente a la operación a invocar.
Este cliente recibirá como respuesta un XML y debe ser capaz de leerlo y extraer de él la información necesaria.
Para programar los clientes de estos Webservices se debe partir de los ***wsdl del Framework de servicios Documentales*** publicados en https://gestiona.madrid.org/soja_int/run/j/EnlaceRecurso.icm?cd_recurso=1122.

4.2.2 GENERACIÓN DE FICHEROS XML DE ENTRADA A LOS SERVICIOS WEB.

Para la creación y manejo de los ficheros xml que se le pasarán como parámetro a cada uno de los servicios web se proporciona la librería **docu_util_lib.jar** que mediante el uso de JAXB realiza las operaciones necesarias para el manejo de información mediante XML. Está publicada en
https://gestiona.madrid.org/soja_int/run/j/EnlaceLibreria.icm?cd_elemento=723
Para generar estos XML se tienen que seguir las especificaciones propuestas por ICM y que se detallan en el **Manual_manejo_XML.pdf** publicado en
https://gestiona.madrid.org/soja_int/run/j/EnlaceManual.icm?cd_elemento=781 .
Estos xml además deben cumplir los esquemas de los mismos, publicados como ***Esquemas del FrameWork de Servidios Documentales*** en https://gestiona.madrid.org/soja_int/run/j/EnlaceRecurso.icm?cd_recurso=1142

4.2.3 URL'S DE ACCESO

Desarrollo: http://icmaplic31b:7002/docu_ws/services

4.2.4 USO DEL SERVICIO WEB DE ADMINISTRACION DE GRUPOS

Este Servicio permite crear y modificar grupos. La forma de llamar a este Servicio Web es llamar al servicio **AdministraconGrupos** en su operación **administrarGrupo,** pasándole como parámetro un xml.

4.2.4.1 DESCRIPCION DEL METODO

Permite crear y modificar grupos.

String administrarGrupo(String strXml)

- **Parámetros de entrada:**

Parámetro	Descripción	Tipo	Obligatorio
strXml	Xml de entrada, según schema **AdministracionGrupos.xsd**	String	SI

- **Retorno:**

String - XML con la respuesta de la operación, indicando si la operación ha sido correcta o no. Contiene dos parámetros: resultado y mensaje. Si el valor del parámetro resultado es 0, indica que no ha habido ningún problema, en caso de tener otro valor, indica que se ha producido un error, descrito en el valor del parámetro mensaje. Este xml sigue el schema **AdministracionGrupos.xsd**

XML DE RETORNO

```
<?xml version="1.0" encoding="UTF-8"?>
<content>
    <response>
        <resultado>0</resultado>
        <mensaje>String</mensaje>
    </response>
</content>
```

- **Schema AdministracionGrupos.xsd**

Schema AdministracionGrupos.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XMLSpy v2007 (http://www.altova.com) by UF787648 (EMBRACE) -->
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
attributeFormDefault="unqualified">
    <xs:element name="datos_validacion">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="usuario_valid" type="xs:string"/>
                <xs:element name="psw_validacion" type="xs:string"/>
                <xs:element name="docbase" type="xs:string"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="response">
        <xs:complexType>
            <xs:all>
                <xs:element ref="resultado"/>
                <xs:element ref="mensaje"/>
            </xs:all>
            <xs:attribute name="cerrar_sesion" type="xs:boolean"/>
            <xs:attribute name="sesion" type="xs:string"/>
        </xs:complexType>
    </xs:element>
```

```
<xs:element name="grupo">
        <xs:complexType>
            <xs:attribute name="nombre" type="xs:string"/>
            <xs:attribute name="accion" type="xs:string"/>
        </xs:complexType>
    </xs:element>
    <xs:element name="lista_grupos">
        <xs:complexType>
            <xs:sequence>
                <xs:element ref="grupo" minOccurs="0" maxOccurs="unbounded"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="usuario">
        <xs:complexType>
<xs:attribute name="nombre" type="xs:string"/>
            <xs:attribute name="accion" type="xs:string"/>
        </xs:complexType>
    </xs:element>
    <xs:element name="lista_usuarios">
        <xs:complexType>
            <xs:sequence>
                <xs:element ref="usuario" minOccurs="0" maxOccurs="unbounded"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="content">
        <xs:complexType>
            <xs:choice>
                <xs:element ref="request"/>
                <xs:element ref="response"/>
            </xs:choice>
        </xs:complexType>
    </xs:element>
<xs:element name="request">
        <xs:complexType>
            <xs:all>
                <xs:element ref="id" minOccurs="0"/>
                <xs:element ref="email" minOccurs="0"/>
                <xs:element ref="descripcion"/>
                <xs:element ref="nombre_grupo" minOccurs="0"/>
                <xs:element ref="datos_validacion"/>
                <xs:element ref="lista_usuarios" minOccurs="0"/>
                <xs:element ref="lista_grupos" minOccurs="0"/>
                <xs:element ref="codigo_aplicacion" minOccurs="0"/>
                <xs:element ref="tipo_documental" minOccurs="0"/>
            </xs:all>
            <xs:attribute name="session" type="xs:string"/>
            <xs:attribute name="cerrar_sesion" type="xs:boolean"/>
        </xs:complexType>
</xs:element>
<xs:element name="id" type="xs:string"/>
<xs:element name="email" type="xs:string"/>
<xs:element name="descripcion" type="xs:string"/>
```

```
        <xs:element name="nombre_grupo" type="xs:string"/>
        <xs:element name="codigo_aplicacion" type="xs:string"/>
        <xs:element name="tipo_documental" type="xs:string"/>
        <xs:element name="resultado" type="xs:int"/>
        <xs:element name="mensaje" type="xs:string"/>
    </xs:schema>
```

Indicaciones:

- Los parámetros ***<usuario_valid>, <psw_validacion> y <docbase>*** son necesarios para realizar las validaciones de usuario e iniciar la sesión de documentum
- El parámetro ***<descripcion>*** es obligatorio
- Los parámetros **<id>** y **<nombre_grupo>** son excluyentes.
- Si se desea dar de alta un grupo se utilizatá el parámetro **<nombre_grupo>** donde se informará del nombre del grupo que se quiere crear.
- Para modificar un grupo se incluirá el parámetro **<id>** con el identificador del grupo a modificar y se omitirá **<nombre_grupo>**.

- **Casos de Prueba**

Caso 1: Alta de grupos

```
<?xml version="1.0" encoding="UTF-8"?>
  <content xsi:noNamespaceSchemaLocation="AdministracionGrupos.xsd"
     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <request>
        <email>mtangel@indra.es</email>
        <descripcion>Prueba ws de febrero</descripcion>
      <nombre_grupo>icm_gr_frameEner</nombre_grupo>
        <datos_validacion>
            <usuario_valid>dmadmin</usuario_valid>
            <psw_validacion>dmadmin</psw_validacion>
            <docbase>icm</docbase>
        </datos_validacion>
       <lista_usuarios>
          <usuario accion="add" nombre="icm"/>
       </lista_usuarios>
          <lista_grupos>
          <grupo accion="add" nombre="icm_gr_config"/>
        </lista_grupos>
  </request>
</content>
```

Caso 2: Modificación de un grupo

```
<?xml version="1.0" encoding="UTF-8"?>
        <content xsi:noNamespaceSchemaLocation="AdministracionGrupos.xsd"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
          <request>
            <email>mtangel@indra.es</email>
            <descripcion>Prueba ws de febrero</descripcion>
             <id>1203a56380000900</id>
             <datos_validacion>
                    <usuario_valid>dmadmin</usuario_valid>
                    <psw_validacion>dmadmin</psw_validacion>
                    <docbase>icm</docbase>
              </datos_validacion>
               <lista_usuarios>
                        <usuario accion="add" nombre="icm"/>
               </lista_usuarios>
              <lista_grupos>
                         <grupo accion="add" nombre="icm_gr_config"/>
                </lista_grupos>
          </request>
        </content>
```

4.2.5 USO DEL SERVICIO WEB DE BORRAR DOCUMENTO

Este Servicio permite el borrado del documento que se corresponde con el identificador del documento que se le pasa como parámetro. La forma de llamar a este Servicio Web es llamar al servicio **BorrarDocumento** en su operación **borrarDocumento,** pasándole como parámetro un xml.

4.2.5.1 DESCRIPCION DEL METODO

Permite borrar documentos

String borrarDocumento(String strXml)

- **Parámetros de entrada:**

Parámetro	Descripción	Tipo	Obligatorio
strXml	Xml de entrada, según schema **BorrarDocumento.xsd**, con los datos del documento a borrar y del usuario que lleva a cabo la operación	String	SI

- **Retorno:**

String - XML con la respuesta de la operación, indicando si la operación ha sido correcta o no. Contiene dos parámetros: resultado y mensaje. Si el valor del parámetro resultado es 0, indica que no ha habido ningún

problema, en caso de tener otro valor, indica que se ha producido un error, descrito en el valor del parámetro mensaje. Este xml sigue el schema **BorrarDocumento.xsd**.

XML DE RETORNO

```
<?xml version="1.0" encoding="UTF-8"?>
<content>
    <response>
        <resultado>0</resultado>
        <mensaje>String</mensaje>
    </response>
</content>
```

- **Schema BorrarDocumento.xsd**

Schema BorrarDocumento.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XMLSpy v2007 (http://www.altova.com) by UF787648 (EMBRACE) -->
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
attributeFormDefault="unqualified">
    <xs:element name="request">
        <xs:complexType>
            <xs:all>
                <xs:element ref="datos_validacion"/>
                <xs:element ref="id_documento"/>
                <xs:element ref="version"/>
            </xs:all>
            <xs:attribute name="session" type="xs:string"/>
            <xs:attribute name="cerrar_sesion" type="xs:boolean"/>
        </xs:complexType>
    </xs:element>
    <xs:element name="datos_validacion">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="usuario_valid" type="xs:string"/>
                <xs:element name="psw_validacion" type="xs:string"/>
                <xs:element name="docbase" type="xs:string"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="response">
        <xs:complexType>
            <xs:all>
                <xs:element ref="resultado"/>
                <xs:element ref="mensaje"/>
                <xs:element ref="nombre_documento" minOccurs="0"/>
            </xs:all>
            <xs:attribute name="cerrar_sesion" type="xs:boolean"/>
            <xs:attribute name="session" type="xs:string"/>
        </xs:complexType>
    </xs:element>
```

```
  <xs:element name="content">
        <xs:complexType>
            <xs:choice>
                <xs:element ref="request"/>
                <xs:element ref="response"/>
            </xs:choice>
        </xs:complexType>
    </xs:element>
    <xs:element name="id_documento" type="xs:string"/>
    <xs:element name="resultado" type="xs:string"/>
    <xs:element name="mensaje" type="xs:string"/>
    <xs:element name="nombre_documento"/>
    <xs:element name="version" type="xs:string"/>
</xs:schema>
```

Indicaciones:

- Los parámetros ***<usuario_valid>, <psw_validacion> y <docbase>*** son necesarios para realizar las validaciones de usuario e iniciar la sesión de documentum,
- Los parámetros **<version>** y **<id_documento>** son obligatorios
- En el parámetro **<version>**, se indicará si se quieren borrar todas las versiones de un documento o no. Su valor por defecto es **CURRENT,** con este valor se borrará el documento indicado en el parámetro **<id_documento>**
- Para borrar todas las versiones se indicará con la palabra **ALL**
- Para borrar una versión especifica se indica con el identificador del documento, y la palabra **CURRENT** en el parámetro versión.

- **Casos de Prueba**

Caso 1: Borrado de la versión concreta de un documento. Versión igual a CURRENT

```
<?xml version="1.0" encoding="UTF-8"?>
<content xsi:noNamespaceSchemaLocation=" BorrarDocumento.xsd " xmlns:xsi="http://ww
w.w3.org/2001/XMLSchema-instance">
      <request>
          <version>current</version>
           <datos_validacion>
                <usuario_valid>dmadmin</usuario_valid>
                <psw_validacion>dmadmin</psw_validacion>
                <docbase>ICM</docbase>
           </datos_validacion>
           <id_documento>0900177280004ba2</id_documento>
           </request>
</content>
```

Caso 2: Borrado de todas la versiones de un documento. Versión igual a ALL

```
<?xml version="1.0" encoding="UTF-8"?>
<content xsi:noNamespaceSchemaLocation=" BorrarDocumento.xsd " xmlns:xsi="http://ww
w.w3.org/2001/XMLSchema-instance">
    <request>
        <version>all</version>
        <datos_validacion>
            <usuario_valid>dmadmin</usuario_valid>
            <psw_validacion>dmadmin</psw_validacion>
            <docbase>ICM</docbase>
        </datos_validacion>
        <id_documento>0900177280004ba2</id_documento>
    </request>
</content>
```

4.2.6 USO DEL SERVICIO WEB DE PEDIR RENDITION

Este Servicio permite generar transformaciones de documentos a formatos .pdf y .html. La forma de llamar a este Web Service es llamar al servicio PedirRendition en su operación **pedirRendition**, a la cual se le pasará el identificador del documento a transformar y el formato deseado como parámetros en un xml

4.2.6.1 DESCRIPCION DEL METODO

Permite generar transformaciones de documentos a formatos .pdf y .html

String pedirRendition(String strXml)

- **Parámetros de entrada:**

Parámetro	Descripción	Tipo	Obligatorio
strXml	Xml de entrada, según schema **PedirRendition.xsd**	String	SI

- **Retorno:**

String - XML con la respuesta de la operación, indicando si la operación ha sido correcta o no. Contiene dos parámetros: resultado y mensaje. Si el valor del parámetro resultado es 0, indica que no ha habido ningún problema, en caso de tener otro valor, indica que se ha producido un error, descrito en el valor del parámetro mensaje. Este xml sigue el schema **PedirRendition.xsd**

XML DE RETORNO

```
<?xml version="1.0" encoding="UTF-8"?>
<content>
    <response>
        <resultado>0</resultado>
        <mensaje>String</mensaje>
    </response>
</content>
```

- **Schema PedirRendition.xsd**

Schema PedirRendition.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XMLSpy v2007 (http://www.altova.com) by UF787648 (EMBRACE) -->
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
attributeFormDefault="unqualified">
    <xs:element name="request">
        <xs:complexType>
            <xs:all>
                <xs:element ref="datos_validacion"/>
                <xs:element ref="id_documento"/>
                <xs:element ref="formato"/>
            </xs:all>
            <xs:attribute name="session" type="xs:string"/>
            <xs:attribute name="cerrar_sesion" type="xs:boolean"/>
        </xs:complexType>
    </xs:element>
    <xs:element name="datos_validacion">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="usuario_valid" type="xs:string"/>
                <xs:element name="psw_validacion" type="xs:string"/>
                <xs:element name="docbase" type="xs:string"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="response">
        <xs:complexType>
            <xs:all>
                <xs:element ref="resultado"/>
                <xs:element ref="mensaje"/>
            </xs:all>
            <xs:attribute name="cerrar_sesion" type="xs:boolean"/>
            <xs:attribute name="session" type="xs:string"/>
        </xs:complexType>
    </xs:element>
```

```
    <xs:element name="content">
        <xs:complexType>
            <xs:choice>
                <xs:element ref="request"/>
                <xs:element ref="response"/>
            </xs:choice>
        </xs:complexType>
    </xs:element>
     <xs:element name="id_documento" type="xs:string"/>
    <xs:element name="resultado" type="xs:string"/>
    <xs:element name="mensaje" type="xs:string"/>
    <xs:element name="formato" type="xs:string"/>
</xs:schema>
```

Indicaciones:

- Los parámetros ***<usuario_valid>, <psw_validacion> y <docbase>*** son necesarios para realizar las validaciones de usuario e iniciar la sesión de documentum,
- El parámetro <id_documento> es obligatorio
- Los posibles valores para el parámetro <formato> son PDF o HTML

- **Casos de Prueba**

Caso 1: Petición de rendition. Todos los datos correctos. Formato pdf

```
<?xml version="1.0" encoding="UTF-8"?>
<content xsi:noNamespaceSchemaLocation="pedirRendition.xsd" xmlns:xsi="http://ww
w.w3.org/2001/XMLSchema-instance">
    <request>
        <datos_validacion>
            <usuario_valid>dmadmin</usuario_valid>
            <psw_validacion>dmadmin</psw_validacion>
            <docbase>ICM</docbase>
        </datos_validacion>
        <formato>pdf</formato>
        <id_documento>09001772800053b7</id_documento>
    </request>
</content>
```

Caso 2: Petición de rendition. Todos los datos correctos. Formato html

```
<?xml version="1.0" encoding="UTF-8"?>
<content xsi:noNamespaceSchemaLocation="pedirRendition.xsd" xmlns:xsi="http://ww
w.w3.org/2001/XMLSchema-instance">
    <request>
        <datos_validacion>
            <usuario_valid>dmadmin</usuario_valid>
            <psw_validacion>dmadmin</psw_validacion>
            <usuario>dmadmin</usuario>
            <docbase>ICM</docbase>
        </datos_validacion>
        <formato>html</formato>
        <id_documento>09001772800053b7</id_documento>
    </request>
</content>
```

4.2.7 USO DEL SERVICIO WEB DE VER DOCUMENTO

Este Servicio permite la visualización del documento que se corresponde con el identificador del documento que se le pasa como parámetro en un xml. El retorno de este WebService es un objeto en el cual en su primera posición está el Datahandler del contenido del documento, que el cliente se encargará de transformar en un fichero físico para guardar en local y visualizarlo, y en la segunda posición devuelve un xml en el que nos indica el nombre del documento consultado. La forma de llamar a este Web Service es llamar al servicio VerDocumento en su operación **verDocumento.**

4.2.7.1 DESCRIPCION DEL METODO

Permite visualizar documentos

Object[] verDocumento(String strXml)

- **Parámetros de entrada:**

Parámetro	Descripción	Tipo	Obligatorio
strXml	Xml de entrada, según schema **VerDocumento.xsd**	String	SI

- **Retorno:**

Object[ ] – objeto, en el que en su primera posición se encuentra el DataHandler del documento que se quiere visualizar y que el cliente transformará en un fichero físico que guardará en local y visualizará, la segunda posición del objeto estará informada con un XML que contiene la respuesta de la operación, indicando si la operacion ha sido correcta o no. Contiene tres parámetros: nombre_documento, resultado y mensaje. Si el valor del parámetro resultado es 0, indica que no ha habido ningún problema, en caso de tener otro valor, indica que se ha producido

un error, descrito en el valor del parámetro mensaje. El parámetro *nombre_documento* contiene el nombre del documento. Este xml sigue el schema **VerDocumento.xsd**

XML DE RETORNO

```
<?xml version="1.0" encoding="UTF-8"?>
<content>
    <response>
        <resultado>0</resultado>
        <mensaje>String</mensaje>
        <nombre_documento>String</nombre_documento>
    </response>
</content>
```

- **Schema VerDocumento.xsd**

Schema VerDocumento.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XMLSpy v2007 (http://www.altova.com) by UF787648 (EMBRACE) -->
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
attributeFormDefault="unqualified">
    <xs:element name="request">
        <xs:complexType>
            <xs:all>
                <xs:element ref="datos_validacion"/>
                <xs:element name="id_documento" type="xs:string"/>
            </xs:all>
            <xs:attribute name="session" type="xs:string"/>
            <xs:attribute name="cerrar_sesion" type="xs:boolean"/>
        </xs:complexType>
    </xs:element>
    <xs:element name="datos_validacion">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="usuario_valid" type="xs:string"/>
                <xs:element name="psw_validacion" type="xs:string"/>
                <xs:element name="docbase" type="xs:string"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
     <xs:element name="response">
        <xs:complexType>
            <xs:all>
                <xs:element name="resultado" type="xs:string"/>
                <xs:element name="mensaje" type="xs:string"/>
                <xs:element name="nombre_documento" type="xs:string" minOccurs="0"/>
            </xs:all>
```

```
            <xs:attribute name="cerrar_sesion" type="xs:boolean"/>
            <xs:attribute name="session" type="xs:string"/>
        </xs:complexType>
    </xs:element>
    <xs:element name="content">
        <xs:complexType>
            <xs:choice>
                <xs:element ref="request"/>
                <xs:element ref="response"/>
            </xs:choice>
        </xs:complexType>
    </xs:element>
</xs:schema>
```

Indicaciones:

- Los parámetros ***<usuario_valid>, <psw_validacion> y <docbase>*** son necesarios para realizar las validaciones de usuario e iniciar la sesión de documentum,
- El parámetro **<id_documento>** es obligatorio.
- El resto de parámetros son opcionales.

- **Casos de Prueba**

Caso 1 –Visualización de documento.

```
<?xml version="1.0" encoding="UTF-8"?>
<content xsi:noNamespaceSchemaLocation="verdocumento.xsd" xmlns:xsi="http://ww
w.w3.org/2001/XMLSchema-instance">
    <request>
        <datos_validacion>
            <usuario_valid>dmadmin</usuario_valid>
            <psw_validacion>dmadmin</psw_validacion>
            <docbase>ICM</docbase>
        </datos_validacion>
        <id_documento>0900177280004ba2</id_documento>
    </request>
</content>
```

4.2.8 USO DEL SERVICIO WEB DE TABLAS EXTERNAS

Este WebService permite realizar modificaciones, inserciones y borrados en tablas externas. La forma de llamar a este Web Service es llamar al servicio **GestionTablas** en su operación **gestionTablas,** pasándole como parámetro un xml.

4.2.8.1 DESCRIPCION DEL METODO

Permite realizar modificaciones, inserciones y borrados en tablas externas

String gestionTablas(String strXml)

- **Parámetros de entrada:**

Parámetro	Descripción	Tipo	Obligatorio
strXml	Xml de entrada, según schema **GestionTablas.xsd**	String	SI

- **Retorno:**

String - XML con la respuesta de la operación, indicando si la operacion ha sido correcta o no. Contiene dos parámetros: resultado y mensaje. Si el valor del parámetro resultado es 0, indica que no ha habido ningún problema, en caso de tener otro valor, indica que se ha producido un error, descrito en el valor del parámetro mensaje. Este xml sigue el schema **GestionTablas.xsd**

XML DE RETORNO

```
<?xml version="1.0" encoding="UTF-8"?>
<content>
    <response>
        <resultado>0</resultado>
        <mensaje>String</mensaje>
    </response>
</content>
```

- **Schema GestionTablas.xsd**

Schema GestionTablas.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XMLSpy v2007 (http://www.altova.com) by UF787648 (EMBRACE) -->
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
attributeFormDefault="unqualified">
    <xs:element name="datos_validacion">
        <xs:complexType>
            <xs:all>
                <xs:element name="usuario_valid" type="xs:string"/>
                <xs:element name="psw_validacion" type="xs:string"/>
                <xs:element name="docbase" type="xs:string"/>
            </xs:all>
        </xs:complexType>
    </xs:element>
    <xs:element name="response">
        <xs:complexType>
            <xs:all>
                <xs:element name="resultado" type="xs:int"/>
                <xs:element name="mensaje" type="xs:string"/>
            </xs:all>
```

```
        <xs:attribute name="session" type="xs:string"/>
        <xs:attribute name="cerrar_sesion" type="xs:boolean"/>
    </xs:complexType>
</xs:element>
<xs:element name="lista_Atributos">
    <xs:complexType>
        <xs:sequence>
            <xs:element ref="campo" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="lista_Busqueda">
    <xs:complexType>
        <xs:sequence>
            <xs:element ref="campo_busqueda" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="operacion" type="xs:string"/>
<xs:element name="request">
    <xs:complexType>
        <xs:all>
            <xs:element ref="datos_validacion"/>
            <xs:element name="tabla" type="xs:string"/>
            <xs:element ref="operacion"/>
            <xs:element ref="lista_Atributos" minOccurs="0"/>
            <xs:element ref="lista_Busqueda" minOccurs="0"/>
        </xs:all>
        <xs:attribute name="session" type="xs:string"/>
        <xs:attribute name="cerrar_sesion" type="xs:boolean"/>
    </xs:complexType>
</xs:element>
<xs:element name="valor_campo" type="xs:string"/>
<xs:element name="campo_busqueda">
    <xs:complexType>
        <xs:simpleContent>
            <xs:extension base="xs:string">
                <xs:attribute name="nombre" type="xs:string"/>
            </xs:extension>
        </xs:simpleContent>
    </xs:complexType>
</xs:element>
<xs:element name="campo">
    <xs:complexType>
        <xs:simpleContent>
            <xs:extension base="xs:string">
                <xs:attribute name="nombre" type="xs:string"/>
                <xs:attribute name="clave" type="xs:boolean"/>
             </xs:extension>
        </xs:simpleContent>
    </xs:complexType>
</xs:element>
```

```
    <xs:element name="tabla">
        <xs:complexType>
            <xs:all>
                <xs:element name="tabla_relacion" type="xs:string"/>
                <xs:element name="campo_relacion" type="xs:string"/>
                <xs:element name="campo_origen" type="xs:string"/>
            </xs:all>
        </xs:complexType>
    </xs:element>
    <xs:element name="content">
        <xs:complexType>
            <xs:choice>
                <xs:element ref="request"/>
                <xs:element ref="response"/>
            </xs:choice>
     </xs:complexType>
    </xs:element>
</xs:schema>
```

Indicaciones:

- Los parámetros ***<usuario_valid>, <psw_validacion> y <docbase>*** son necesarios para realizar las validaciones de usuario e iniciar la sesión de documentum,
- El parámetro ***<tabla>*** es obligatorio, sirve para indicar el nombre de la tabla sobre la que se realiza la operación.
- El parámetro ***<operacion>*** es obligatorio, sirve para indicar la operación a realizar. Los valores que puede tomar son: **Insert, Delete, Update**.
- El parámetro ***<lista_Atributos>*** es opcional, esta lista se utiliza en el caso de que la operación que se realice sea una inserción o una modificación. En el caso de las modificaciones, contendrá el nuevo valor a asignar (el SET de la sentencia UPDATE). En el caso de una inserción y que la tabla tratada tenga algún tipo de código automático se debe introducir este a 0 si queremos que salte el triggers de la tabla, aunque también podemos introducir de manera manual el mismo. Para hacer dicha operaciones se rellena la etiqueta ***campo*** asignando en ***nombre*** el valor del campo de la tabla y su valor, en la etiqueta clave se especifica si ese campo es clave en la tabla o no.
- El parámetro ***<lista_Busqueda>*** es opcional, esta lista se usa en el caso de que la operación que se vaya a realizar sea un borrado o una modificación. En el caso de las modificaciones, contendrá el valor antiguo que se desea cambiar (el WHERE de la operación UPDATE), en el borrado también se trata de la cláusula WHERE de la query. Para hacer dicha operaciones se rellena la etiqueta ***campo_busqueda*** asignando en ***nombre*** el valor del campo de la tabla y su valor.
- Es importante especificar que en estos parámetros cuando lo que se quiere tratar es un campo string este debe introducirse entre comillas (‘), como se puede ver en los casos de prueba.

- **Casos de Prueba**

Caso 1 – Realizar una inserción en una tabla.

```
<?xml version="1.0" encoding="UTF-8"?>
<content xsi:noNamespaceSchemaLocation="GestionTablas.xsd"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
        <request>
                <datos_validacion>
                        <usuario_valid>dmadmin</usuario_valid>
                        <psw_validacion>dmadmin</psw_validacion>
                        <docbase>ICM</docbase>
                </datos_validacion>
                <tabla>dm_dbo. AUX_TIPO_EXP</tabla>
                <operacion>INSERT</operacion>
                <lista_Atributos>
                        <campo nombre="CODE" clave=”true”>0</campo>
                        <campo nombre="DESCRIPCION" clave=”false”>’descripción ejg’</campo>
                </lista_Atributos>
                <lista_Busqueda></lista_Busqueda>
        </request>
</content>
```

Caso 2 – Realizar un modificacion de un campo de la tabla.

```
<?xml version="1.0" encoding="UTF-8"?>
<content xsi:noNamespaceSchemaLocation="GestionTablas.xsd"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
          <request>
                <datos_validacion>
                        <usuario_valid>dmadmin</usuario_valid>
                        <psw_validacion>dmadmin</psw_validacion>
                        <docbase>ICM</docbase>
                </datos_validacion>
                <tabla>dm_dbo. AUX_TIPO_EXP</tabla>
                <operacion>UPDATE</operacion>
                <lista_Atributos>
                        <campo nombre="DESCRIPCION" clave=”false”>’descripción modificada’ </campo>
                </lista_Atributos>
                <lista_Busqueda>
                <campo_busqueda nombre="DESCRIPCION">’descripción’ </campo_busqueda>
        </lista_Busqueda>
          </request>
</content>
```

Caso 3 – Borrar un campo de la tabla.

```
<?xml version="1.0" encoding="UTF-8"?>
<content xsi:noNamespaceSchemaLocation="GestionTablas.xsd"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
        <request>
            <datos_validacion>
                <usuario_valid>dmadmin</usuario_valid>
                <psw_validacion>dmadmin</psw_validacion>
                <docbase>ICM</docbase>
            </datos_validacion>
            <tabla>dm_dbo. AUX_TIPO_EXP</tabla>
            <operacion>DELETE</operacion>
            <lista_Atributos></lista_Atributos>
            <lista_Busqueda>
            <campo_busqueda nombre="DESCRIPCION">'descripción modificada'</campo_busqueda>
      </lista_Busqueda>
        </request>
</content>
```

4.2.9 USO DEL SERVICIO WEB DE CHECK DOCUMENTO

Este Servicio permite bloquear, desbloquear y registrar los cambios en un documento por un usuario previamente bloqueado por el mismo. La forma de llamar a este Servicio Web es llamar al servicio **CheckDocumento** con alguna de las operaciones siguientes: **checkin, checkout, cancel checkout,** pasándole como parámetro un xml**.**

4.2.9.1 DESCRIPCION DEL METODO CHECKIN

Permite registrar los cambios que ha realizado un usuario en un documento, previamente bloqueado por el mismo

Object[] checkin(String strXml, DataHandler dhFile) throws WSException

- **Parámetros de entrada:**

Parámetro	Descripción	Tipo	Obligatorio
strXml	Xml de entrada, según schema **CheckDocumento.xsd**	String	SI
dhFile	Objeto DataHandler con el contenido del documento modificado que se quiere registrar. Este Datahandler lo crea el cliente a partir del path del documento que se desea añadir como contenido.	DataHandler	NO

- **Retorno:**

Object[ ] – objeto, en el que en su primera posición se encuentra el DataHandler del contenido del documento si en los parámetros de entrada se especifica que se quiere guardar copia del mismo, la segunda posición del objeto estará informada con un XML que contiene la respuesta de la operación, indicando si la operacion ha sido correcta o no. Contiene tres parámetros: nombre_documento, resultado y mensaje. Si el valor del parámetro resultado es 0, indica que no ha habido ningún problema, en caso de tener otro valor, indica que se ha producido un error, descrito en el valor del parámetro mensaje. El parámetro *nombre_documento* contiene el nombre del documento Este xml sigue el schema **CheckDocumento.xsd**

XML DE RETORNO

```
<?xml version="1.0" encoding="UTF-8"?>
<response>
    <resultado>0</resultado>
    <mensaje>String</mensaje>
    <nombre_documento>String</nombre_documento>
</response>
```

- **Schema**

Schema CheckDocumento.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XMLSpy v2007 (http://www.altova.com) by UF787648 (EMBRACE) -->
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
attributeFormDefault="unqualified">
    <xs:element name="request">
        <xs:complexType>
            <xs:all>
                <xs:element ref="datos_validacion"/>
                <xs:element name="id_documento" type="xs:string"/>
                <xs:element name="extension" type="xs:string" minOccurs="0"/>
                <xs:element name="nombreDocumento" type="xs:string" minOccurs="0"/>
                <xs:element name="descripcion" type="xs:string" minOccurs="0"/>
                <xs:element name="locked" type="xs:string" minOccurs="0"/>
                <xs:element name="version" type="xs:string" minOccurs="0"/>
                <xs:element name="labelVersion" type="xs:string" minOccurs="0"/>
                <xs:element name="currentVersion" type="xs:string" minOccurs="0"/>
                <xs:element name="subscribe" type="xs:string" minOccurs="0"/>
                <xs:element name="indexar" type="xs:string" minOccurs="0"/>
            </xs:all>
            <xs:attribute name="session" type="xs:string"/>
            <xs:attribute name="cerrar_sesion" type="xs:boolean"/>
        </xs:complexType>
    </xs:element>
    <xs:element name="datos_validacion">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="usuario_valid" type="xs:string"/>
                <xs:element name="psw_validacion" type="xs:string"/>
                <xs:element name="docbase" type="xs:string"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="response">
        <xs:complexType>
            <xs:all>
                <xs:element name="resultado" type="xs:string"/>
                <xs:element name="mensaje" type="xs:string" minOccurs="0"/>
                <xs:element name="nombre_documento" type="xs:string" minOccurs="0"/>
            </xs:all>
            <xs:attribute name="cerrar_sesion" type="xs:boolean"/>
            <xs:attribute name="session" type="xs:string"/>
        </xs:complexType>
    </xs:element>
```

```
    <xs:element name="content">
        <xs:complexType>
            <xs:choice>
                <xs:element ref="request"/>
                <xs:element ref="response"/>
            </xs:choice>
        </xs:complexType>
    </xs:element>
</xs:schema>
```

Indicaciones:

- Los parámetros ***<usuario_valid>, <psw_validacion> y <docbase>*** son necesarios para realizar las validaciones de usuario e iniciar la sesión de documentum,
- El parámetro **<id_documento>** es obligatorio
- Los siguientes parámetros son opcionales:
  - **<extension>,** parámetro para informar la extensión del nuevo contenido que se quiera dar al documento al hacer su registro
  - **<nombreDocumento>** indica el nombre con el cual el documento es guardado en la base de datos de Documentum, si este parámetro no está informado se quedará con el nombre anterior.
  - **<descripción>**, hace referencia a la descripción con la que se registra el documento.
  - El parámetro **<locked>** si está informado con 'true' significa que el nuevo documento registrado debe permanecer bloqueado por el usuario.
  - Mediante el parámetro **<version>** se indica la nueva versión con la que será registrado el documento, aquí se indicara, 'minor', 'same' (misma version), o 'major' , si esta informado **<labelVersion>** la versión registrado contemplará esta etiqueta, y si **<currentVersion>** está informado a 'true' la etiqueta CURRENT estará en el nuevo documento registrado .
  - **<susbcribe>** e **<indexar>** son parámetro informados a 'true' o 'false' y significan si están informados a 'true' que se subscribirá el documento y que se indexará el texto del documento, respectivamente.

- **Casos de Prueba**

Caso 1: Registrar un documento especificando la ruta. Sin guardar copia

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<content>
  <request>
    <datos_validacion>
        <usuario_valid>dmadmin</usuario_valid>
        <psw_validacion>dmadmin</psw_validacion>
        <docbase>ICM</docbase>
    </datos_validacion>
    <id_documento>090017728000e2e7</id_documento>
    <extension>txt</extension>
    <nombreDocumento>Documento</nombreDocumento>
    <descripcion>Prueba de Checkin</descripcion>
    <locked>false</locked>
   <version>minor</version>
    <labelVersion>Version</labelVersion>
    <currentVersion>true</currentVersion>
    <subscribe>false</subscribe>
    <indexar>false</indexar>
  </request>
</content>
```

Caso 2: Registrar un documento, guardando copia y bloqueandolo.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<content>
  <request>
     <datos_validacion>
          <usuario_valid>dmadmin</usuario_valid>
           <psw_validacion>dmadmin</psw_validacion>
           <docbase>ICM</docbase>
      </datos_validacion>
      <id_documento>090017728000e28b</id_documento>
      <extension>pdf</extension>
      <nombreDocumentoDocumento Checkin</nombreDocumento>
      <descripcion>Prueba checkin dos</descripcion>
      <locked>true</locked>
      <version>major</version>
      <labelVersion>MAG</labelVersion>
      <currentVersion>true</currentVersion>
      <subscribe>false</subscribe>
      <indexar>false</indexar>
   </request>
</content>
```

4.2.9.2 DESCRIPCION DEL METODO CHECKOUT

Permite bloquear un documento

public Object[] checkout(String strXml) throws WSException

- **Parámetros de entrada:**

Parámetro	Descripción	Tipo	Obligatorio
strXml	Xml de entrada, según schema **CheckDocumento.xsd**	String	SI

- **Retorno:**

Object[ ] – objeto, en el que en su primera posición se encuentra el DataHandler del contenido del documento, que el cliente se encargará de transformar en un fichero físico para guardar en local, esta copia en local se hará en un sitio especifico que indique el cliente, por ejemplo :
dirDestino=**new** File(System.*getProperty*("user.home") + System.*getProperty*("file.separator") + "Checkout");
La segunda posición del objeto estará informada con un XML que contiene la respuesta de la operación, indicando si la operacion ha sido correcta o no. Contiene dos parámetros: resultado y mensaje. Si el valor del parámetro resultado es 0, indica que no ha habido ningún problema, en caso de tener otro valor, indica que se ha producido un error, descrito en el valor del parámetro mensaje. Este xml sigue el schema **CheckDocumento.xsd**

XML DE RETORNO

```
<?xml version="1.0" encoding="UTF-8"?>
<response>
    <resultado>0</resultado>
    <mensaje>String</mensaje>
    <nombre_documento>String</nombre_documento>
</response>
```

El parámetro **<nombre_documento>** contiene el nombre del documento

- **Schema**

Schema CheckDocumento.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XMLSpy v2007 (http://www.altova.com) by UF787648 (EMBRACE) -->
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
attributeFormDefault="unqualified">
    <xs:element name="request">
        <xs:complexType>
            <xs:all>
                <xs:element ref="datos_validacion"/>
                <xs:element name="id_documento" type="xs:string"/>
                <xs:element name="extension" type="xs:string" minOccurs="0"/>
                <xs:element name="nombreDocumento" type="xs:string" minOccurs="0"/>
                <xs:element name="descripcion" type="xs:string" minOccurs="0"/>
                <xs:element name="locked" type="xs:string" minOccurs="0"/>
                <xs:element name="version" type="xs:string" minOccurs="0"/>
                <xs:element name="labelVersion" type="xs:string" minOccurs="0"/>
                <xs:element name="currentVersion" type="xs:string" minOccurs="0"/>
                <xs:element name="subscribe" type="xs:string" minOccurs="0"/>
                <xs:element name="indexar" type="xs:string" minOccurs="0"/>
            </xs:all>
            <xs:attribute name="session" type="xs:string"/>
            <xs:attribute name="cerrar_sesion" type="xs:boolean"/>
        </xs:complexType>
    </xs:element>
    <xs:element name="datos_validacion">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="usuario_valid" type="xs:string"/>
                <xs:element name="psw_validacion" type="xs:string"/>
                <xs:element name="docbase" type="xs:string"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="response">
        <xs:complexType>
            <xs:all>
                <xs:element name="resultado" type="xs:string"/>
                <xs:element name="mensaje" type="xs:string" minOccurs="0"/>
                <xs:element name="nombre_documento" type="xs:string" minOccurs="0"/>
            </xs:all>
            <xs:attribute name="cerrar_sesion" type="xs:boolean"/>
            <xs:attribute name="session" type="xs:string"/>
        </xs:complexType>
    </xs:element>
```

```
    <xs:element name="content">
        <xs:complexType>
            <xs:choice>
                <xs:element ref="request"/>
                <xs:element ref="response"/>
            </xs:choice>
        </xs:complexType>
    </xs:element>
</xs:schema>
```

Indicaciones:

- Los parámetros ***<usuario_valid>, <psw_validacion> y <docbase>*** son necesarios para realizar las validaciones de usuario e iniciar la sesión de documentum,
- El parámetro **<id_documento>** es obligatorio

- **Casos de Prueba**

Caso 1: Checkout pasándole identificador del documento

```
<?xml version="1.0" encoding="UTF-8"?>
<content xsi:noNamespaceSchemaLocation="checkout.xsd" xmlns:xsi="http://ww
w.w3.org/2001/XMLSchema-instance">
    <request>
        <datos_validacion>
            <usuario_valid>dmadmin</usuario_valid>
            <psw_validacion>dmadmin</psw_validacion>
            <docbase>ICM</docbase>
        </datos_validacion>
        <id_documento>0900177280002e7e</id_documento>
    </request>
</content>
```

4.2.9.3 DESCRIPCION DEL METODO CANCEL CKECKOUT

Permite desbloquear un documento previamente bloqueado con la operación Checkout

String cancelCheckout(String strXml)

- **Parámetros de entrada:**

Parámetro	Descripción	Tipo	Obligatorio
strXml	Xml de entrada, según schema **CheckDocumento.xsd**	String	SI

- **Retorno:**

String - XML con la respuesta de la operación, indicando si la operación ha sido correcta o no. Contiene dos parámetros: resultado y mensaje. Si el valor del parámetro resultado es 0, indica que no ha habido ningún problema, en caso de tener otro valor, indica que se ha producido un error, descrito en el valor del parámetro mensaje. Este xml sigue el schema **CheckDocumento.xsd**

XML DE RETORNO

```
<?xml version="1.0" encoding="UTF-8"?>
<response>
    <resultado>0</resultado>
    <mensaje>String</mensaje>
    <nombre_documento>String</nombre_documento>
</response>
```

El parámetro **<nombre_documento>** contiene el nombre del documento.
El cliente que llama a esta operación debe borrar la copia de local que ha dejado el cliente que llamó a la operación de Checkout, siempre y cuando ambas estén especificando la misma ruta:

```
dirDestino=new File(System.getProperty("user.home") + System.getProperty("file.separator") + "Checkout" );
```

- **Schema**

Schema CheckDocumento.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XMLSpy v2007 (http://www.altova.com) by UF787648 (EMBRACE) -->
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
attributeFormDefault="unqualified">
    <xs:element name="request">
        <xs:complexType>
            <xs:all>
                <xs:element ref="datos_validacion"/>
                <xs:element name="id_documento" type="xs:string"/>
                <xs:element name="extension" type="xs:string" minOccurs="0"/>
                <xs:element name="nombreDocumento" type="xs:string" minOccurs="0"/>
                <xs:element name="descripcion" type="xs:string" minOccurs="0"/>
                <xs:element name="locked" type="xs:string" minOccurs="0"/>
                <xs:element name="version" type="xs:string" minOccurs="0"/>
                <xs:element name="labelVersion" type="xs:string" minOccurs="0"/>
                <xs:element name="currentVersion" type="xs:string" minOccurs="0"/>
                <xs:element name="subscribe" type="xs:string" minOccurs="0"/>
                <xs:element name="indexar" type="xs:string" minOccurs="0"/>
            </xs:all>
            <xs:attribute name="session" type="xs:string"/>
            <xs:attribute name="cerrar_sesion" type="xs:boolean"/>
        </xs:complexType>
    </xs:element>
    <xs:element name="datos_validacion">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="usuario_valid" type="xs:string"/>
                <xs:element name="psw_validacion" type="xs:string"/>
                <xs:element name="docbase" type="xs:string"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
```

```
<xs:element name="response">
        <xs:complexType>
            <xs:all>
                <xs:element name="resultado" type="xs:string"/>
                <xs:element name="mensaje" type="xs:string" minOccurs="0"/>
                <xs:element name="nombre_documento" type="xs:string" minOccurs="0"/>
            </xs:all>
            <xs:attribute name="cerrar_sesion" type="xs:boolean"/>
            <xs:attribute name="session" type="xs:string"/>
        </xs:complexType>
</xs:element>
<xs:element name="content">
        <xs:complexType>
            <xs:choice>
                <xs:element ref="request"/>
                <xs:element ref="response"/>
            </xs:choice>
        </xs:complexType>
    </xs:element>
</xs:schema>
```

Indicaciones:

- Los parámetros ***<usuario_valid>, <psw_validacion> y <docbase>*** son necesarios para realizar las validaciones de usuario e iniciar la sesión de documentum,
- El parámetro **<id_documento>** es obligatorio, se corresponde con el identificador del documento que se desea desbloquear

- **Casos de Prueba**

Caso 1: Cancelar Checkout pasándole identificador del documento

```
<?xml version="1.0" encoding="UTF-8"?>
<content xsi:noNamespaceSchemaLocation="cancelarcheckout.xsd" xmlns:xsi="http://ww
w.w3.org/2001/XMLSchema-instance">
    <request>
        <datos_validacion>
            <usuario_valid>dmadmin</usuario_valid>
            <psw_validacion>dmadmin</psw_validacion>
            <docbase>ICM</docbase>
        </datos_validacion>
        <id_documento>09001772800028fe</id_documento>
    </request>
</content>
```

4.2.10 USO DEL SERVICIO WEB DE IMPORTAR DOCUMENTO

Este WebService permite importar y modificar documentos. Permite modificar tanto metadatos como contenido. La forma de llamar a este Servicio Web es llamar al servicio **ImportarModificar** en su operación **ImportarModificarDoc,** pasándole como parámetro un xml

4.2.10.1 DESCRIPCION DEL METODO

Permite importar y modificar documentos

String ImportarModificarDoc(String strXml, DataHandler dhFile) throws WSException

- **Parámetros de entrada:**

Parámetro	Descripción	Tipo	Obligatorio
strXml	Xml de entrada, según schema **ImportarModificar.xsd**	String	SI
dhFile	Objeto DataHandler con el contenido del documento a importar o modificar. Este Datahandler lo crea el cliente a partir del path del documento que se desea añadir como contenido.	DataHandler	NO

- **Retorno:**

String - XML con la respuesta de la operación, indicando si la operacion ha sido correcta o no. Contiene dos parámetros: resultado y mensaje. Si el valor del parámetro resultado es 0, indica que no ha habido ningún problema, en caso de tener otro valor, indica que se ha producido un error, descrito en el valor del parámetro mensaje. En el parámetro idDocumento devuelve el id del documento importado. Este xml sigue el schema **ImportarModificar.xsd**

XML DE RETORNO

```
<?xml version="1.0" encoding="UTF-8"?>
<content>
    <response>
        <idDocumento>String</idDocumento>
        <resultado>0</resultado>
        <mensaje>String</mensaje>
    </response>
</content>
```

- **Schema**

Schema ImportarModificar.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XMLSpy v2007 (http://www.altova.com) by UF787648 (EMBRACE) -->
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
attributeFormDefault="unqualified">
    <xs:element name="response">
        <xs:complexType>
            <xs:all>
                <xs:element ref="mensaje"/>
                <xs:element ref="resultado"/>
                <xs:element ref="idDocumento" minOccurs="0"/>
            </xs:all>
            <xs:attribute name="sesion" type="xs:string"/>
```

```
<xs:attribute name="cerrarSesion" type="xs:boolean"/>
        </xs:complexType>
    </xs:element>
    <xs:element name="docBase" type="xs:string"/>
    <xs:element name="pswValid" type="xs:string"/>
    <xs:element name="usuarioValid" type="xs:string"/>
    <xs:element name="datosValidacion">
        <xs:complexType>
            <xs:all>
                <xs:element ref="usuarioValid"/>
                <xs:element ref="pswValid"/>
                <xs:element ref="docBase"/>
            </xs:all>
        </xs:complexType>
    </xs:element>
    <xs:element name="atributo">
        <xs:complexType>
            <xs:simpleContent>
                <xs:extension base="xs:string">
                    <xs:attribute name="nombre" type="xs:string"/>
                    <xs:attribute name="valorRepeating" type="xs:string"/>
                    <xs:attribute name="operacion" type="xs:string"/>
                    <xs:attribute name="indiceRepeating" type="xs:string"/>
                </xs:extension>
            </xs:simpleContent>
        </xs:complexType>
    </xs:element>
    <xs:element name="extension" type="xs:string"/>
    <xs:element name="listaAtributos">
        <xs:complexType>
            <xs:sequence>
                <xs:element ref="atributo" maxOccurs="unbounded"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="mensaje" type="xs:string"/>
    <xs:element name="resultado" type="xs:int"/>
    <xs:element name="idDocumento" type="xs:string"/>
    <xs:element name="request">
        <xs:complexType>
            <xs:all>
                <xs:element ref="datosValidacion"/>
                <xs:element ref="extension" minOccurs="0"/>
                <xs:element ref="tipoDocumental"/>
                <xs:element ref="rutaRepositorio" minOccurs="0"/>
                <xs:element ref="listaAtributos"/>
                <xs:element ref="acl" minOccurs="0"/>
                <xs:element ref="idDocumento" minOccurs="0"/>
            </xs:all>
            <xs:attribute name="sesion" type="xs:string"/>
            <xs:attribute name="cerrarSesion" type="xs:boolean"/>
        </xs:complexType>
    </xs:element>
    <xs:element name="tipoDocumental" type="xs:string"/>
    <xs:element name="rutaRepositorio" type="xs:string"/>
    <xs:element name="acl">
        <xs:complexType>
            <xs:simpleContent>
                <xs:extension base="xs:string">
                    <xs:attribute name="nombre" type="xs:string"/>
                    <xs:attribute name="dominio" type="xs:string"/>
                </xs:extension>
```

```
            </xs:simpleContent>
        </xs:complexType>
    </xs:element>
    <xs:element name="content">
        <xs:annotation>
            <xs:documentation>Comment describing your root element</xs:documentation>
        </xs:annotation>
        <xs:complexType>
            <xs:choice>
                <xs:element ref="request"/>
                <xs:element ref="response"/>
            </xs:choice>
        </xs:complexType>
    </xs:element>
</xs:schema>
```

Indicaciones:

- Los parámetros ***<usuario_valid>, <psw_validacion> y <docbase>*** son necesarios para realizar las validaciones de usuario e iniciar la sesión de documentum,
- El parámetro **<tipo documental>**, es obligatorio. Para indicar el tipo documental en el que va a tener lugar la importación o la modificación.
- El parámetro **<listaAtributos>**, es obligatorio. En este parámetro se irán informando de los distintos atributos, el campo **operacion** sólo permite los valores **ADD, MODIFY o REMOVE**, los campos **valorRepeating** e **indiceRepeating** son excluyentes, el servicio comprueba siempre el segundo y en caso de estar informado no evalúa el primero. El web service comprueba los distintos atributos de cada tipo documental en el tipo documental, dm_type.
- Los siguientes parámetros son opcionales:
  - **<idDocumento>**, estará informado solo si se trata de una modificación.
  - **<acl>** parámetro en el que se especifican los permisos del documento, si no se especifica nada, por defecto el documento tendrá los permisos de la carpeta.
  - **<ruta_repositorio>** para indicar la ruta de carpetas donde se importará el documento o donde se haya el documento a modificar.
  - **<extensión>** para indicar la extensión del documento a importar. El documento a importar se informa como un Datahandler al servicio, dicho Datahandler será creado por el cliente.

- **Casos de Prueba**

Caso 1: Importación de un documento con contenido

```
<?xml version="1.0" encoding="UTF-8"?>
<contentxsi:noNamespaceSchemaLocation="ImportarModificar.xsd"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
        <request>
                <datosValidacion>
                        <usuarioValid>dmadmin</usuarioValid>
                        <pswValid>dmadmin</pswValid>
                        <docBase>ICM</docBase>
                </datosValidacion>
                <tipoDocumental>cm_doc_general</tipoDocumental>
                <rutaRepositorio>/Cabinet1</rutaRepositorio>
                <extension>pdf</extension>
                <listaAtributos>
                        <atributo nombre="object_name">calendario</atributo>
                        <atributo nombre="periodo_conservacion_fec">calendario</atributo>
                </listaAtributos>
```

```
    </request>
</content>
```

Caso 2: Borrado de todas la versiones de un documento. Versión igual a ALL

```
<?xml version="1.0" encoding="UTF-8"?>
<contentxsi:noNamespaceSchemaLocation="ImportarModificar.xsd"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
        <request>
                <datosValidacion>
                        <usuarioValid>dmadmin</usuarioValid>
                        <pswValid>dmadmin</pswValid>
                        <docBase>ICM</docBase>
                </datosValidacion>
                <tipoDocumental>cm_doc_general</tipoDocumental>
                <rutaRepositorio> </rutaRepositorio>
                <extension>doc </extension>
                  <idDocumento>0093000082457854</idDocumento>
                <listaAtributos>
                        <atributo nombre="object_name">calendario modificacdo</atributo>
                        <atributo nombre="periodo_conservacion_fec">calendario mod</atributo>
                         <atributo nombre="atr_repeating" operación=”MODIFY” valorRepeating=”valor1”
>valorNuevo </atributo>
                </listaAtributos>
        </request>
</content>
```

4.2.11 USO DEL SERVICIO WEB DE BUSCAR DOCUMENTOS

Este Servicio permite realizar consultas en documentum. La forma de llamar a este Servicio Web es llamar al servicio **BuscarDocumentos** en su operación **buscarDocumentos,** pasándole como parámetro un xml.

4.2.11.1 DESCRIPCION DEL METODO

Permite realizar consultas en documentum.

String buscarDocumentos(String strXml)

- **Parámetros de entrada:**

Parámetro	Descripción	Tipo	Obligatorio
strXml	Xml de entrada, según schema **BuscarDocumento.xsd**	String	SI

- **Retorno:**

String - XML con la respuesta de la operación, indicando si la operacion ha sido correcta o no. Contiene dos parámetros: resultado y mensaje. Si el valor del parámetro resultado es 0, indica que no ha habido ningún problema, en caso de tener otro valor, indica que se ha producido un error, descrito en el valor del parámetro mensaje. En el parámetro lista_fila, se genera una línea por cada fila devuelta con todos los atributos solicitados, en el caso de que no exista ningún documento devolverá como resultado 0. Este xml sigue el schema **BuscarDocumento.xsd**

XML DE RETORNO

```
<?xml version="1.0" encoding="UTF-8"?>
<response>
        <resultado_consulta>
                <lista_fila>
                        <columna value='String' nombre='String'/>
                </lista_fila>
                <resultado>0</resultado>

                <mensaje>String</mensaje>

        </resultado_consulta>

</response>
```

- **Schema**

Schema BuscarDocumento.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XMLSpy v2007 (http://www.altova.com) by UF787648 (EMBRACE) -->
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
attributeFormDefault="unqualified">
    <xs:element name="columna">
        <xs:complexType>
            <xs:attribute name="value" type="xs:string"/>
            <xs:attribute name="nombre" type="xs:string"/>
        </xs:complexType>
    </xs:element>
    <xs:element name="lista_fila">
        <xs:complexType>
            <xs:sequence>
                <xs:element ref="columna" minOccurs="0" maxOccurs="unbounded"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="resultado_consulta">
        <xs:complexType>
            <xs:sequence>
                <xs:element ref="lista_fila" minOccurs="0" maxOccurs="unbounded"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
```

```
<xs:element name="content">
        <xs:complexType>
            <xs:choice>
                <xs:element ref="request"/>
                <xs:element ref="response"/>
            </xs:choice>
        </xs:complexType>
    </xs:element>
    <xs:element name="datos_validacion">
        <xs:complexType>
            <xs:all>
                <xs:element name="usuario_valid" type="xs:string"/>
                <xs:element name="psw_validacion" type="xs:string"/>
                <xs:element name="docbase" type="xs:string"/>
            </xs:all>
        </xs:complexType>
    </xs:element>
    <xs:element name="response">
        <xs:complexType>
            <xs:all>
                <xs:element ref="resultado_consulta" minOccurs="0"/>
                <xs:element name="resultado" type="xs:int"/>
                <xs:element name="mensaje" type="xs:string" minOccurs="0"/>
            </xs:all>
            <xs:attribute name="cerrar_session" type="xs:boolean"/>
            <xs:attribute name="session" type="xs:string"/>
        </xs:complexType>
    </xs:element>
    <xs:element name="request">
        <xs:complexType>
            <xs:all>
                <xs:element ref="datos_validacion"/>
                <xs:element name="dql" type="xs:string"/>
            </xs:all>
            <xs:attribute name="session" type="xs:string"/>

            <xs:attribute name="cerrar_sesion" type="xs:boolean"/>
        </xs:complexType>
    </xs:element>
</xs:schema>
```

Indicaciones:

- Los parámetros ***<usuario_valid>, <psw_validacion> y <docbase>*** son necesarios para realizar las validaciones de usuario e iniciar la sesión de documentum,
- El parámetro **<dql>**es obligatorio, sirve para indicar la dql que se quiere ejecutar en la base de datos.

- **Casos de Prueba**

Caso 1: Consulta a una tabla

```
<?xml version="1.0" encoding="UTF-8"?>
<contentxsi:noNamespaceSchemaLocation="BusquedaAvanzada.xsd"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
        <request>
                <datosValidacion>
                        <usuarioValid>dmadmin</usuarioValid>
                        <pswValid>dmadmin</pswValid>
                        <docBase>ICM</docBase>
                </datosValidacion>
                <dql>select * from cm_doc_general</dql>
        </request>
  </content>
```

4.2.12 USO DEL SERVICIO WEB DE MODIFICAR PERMISOS DE UN DOCUMENTO

Este Servicio permite modificar los distintos permisos de los usuarios y grupos asociados a un determinado documento, lo que realmente se hace es cambiar la Acl asociada de dicho documento. Si cuando se va a modificar los permisos de un documento este contiene una acl estática se cambiará la acl del documento. Si el documento tiene una acl dinámica se cambiarán los permisos de la acl, manteniendo el documento esta acl. La forma de llamar a este Servicio Web es llamar al servicio **PermisosDocumento** en su operación **permisosDocumento,** pasándole como parámetro un xml

4.2.12.1 DESCRIPCION DEL METODO

Permite modificar los distintos permisos de los usuarios y grupos asociados a un determinado documento.

String permisosDocumento(String strXml)

- **Parámetros de entrada:**

Parámetro	Descripción	Tipo	Obligatorio
strXml	Xml de entrada, según schema **PermisosDocumento.xsd**	String	SI

- **Retorno:**

String - XML con la respuesta de la operación, indicando si la operacion ha sido correcta o no. Contiene dos parámetros: resultado y mensaje. Si el valor del parámetro resultado es 0, indica que no ha habido ningún problema, en caso de tener otro valor, indica que se ha producido un error, descrito en el valor del parámetro mensaje. Este xml sigue el schema **PermisosDocumento.xsd.**

XML DE RETORNO

```
<?xml version="1.0" encoding="UTF-8"?>
<content>
    <response>
        <resultado>0</resultado>
        <mensaje>String</mensaje>
    </response>
</content>
```

- **Schema**

Schema PermisosDocumento.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XMLSpy v2007 (http://www.altova.com) by UF787648 (EMBRACE) -->
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
attributeFormDefault="unqualified">
    <xs:element name="response">
        <xs:complexType>
            <xs:all>
                <xs:element name="resultado" type="xs:string"/>
                <xs:element name="mensaje" type="xs:string" minOccurs="0"/>
            </xs:all>
            <xs:attribute name="session" type="xs:string"/>
            <xs:attribute name="cerrar_sesion" type="xs:boolean"/>
        </xs:complexType>
    </xs:element>
    <xs:element name="content">
        <xs:complexType>
            <xs:choice>
                <xs:element ref="request"/>
                <xs:element ref="response"/>
            </xs:choice>
        </xs:complexType>
    </xs:element>
    <xs:element name="datos_validacion">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="usuario_valid" type="xs:string"/>
                <xs:element name="psw_validacion" type="xs:string"/>
                <xs:element name="docbase" type="xs:string"/>
            </xs:sequence>
        </xs:complexType>
     </xs:element>
    <xs:element name="permisos">
        <xs:complexType>
            <xs:sequence>
                <xs:element ref="usuario_documento" minOccurs="0" maxOccurs="unbounded"/>
            </xs:sequence>
        </xs:complexType>
```

```
    </xs:element>
    <xs:element name="request">
        <xs:complexType>
            <xs:all>
                <xs:element ref="datos_validacion"/>
                <xs:element name="id_documento" type="xs:string"/>
                <xs:element ref="permisos" minOccurs="0"/>
                <xs:element ref="restricciones" minOccurs="0"/>
            </xs:all>
            <xs:attribute name="session" type="xs:string"/>
            <xs:attribute name="cerrar_sesion" type="xs:boolean"/>
        </xs:complexType>
    </xs:element>
    <xs:element name="usuario_documento">
        <xs:complexType>
            <xs:attribute name="nombre"/>
            <xs:attribute name="permiso"/>
            <xs:attribute name="permiso_extendido"/>
        </xs:complexType>
    </xs:element>
    <xs:element name="restricciones">
        <xs:complexType>
            <xs:sequence>
                <xs:element ref="usuario_documento" minOccurs="0" maxOccurs="unbounded"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
</xs:schema>
```

Indicaciones:

- Los parámetros ***<usuario_valid>, <psw_validacion> y <docbase>*** son necesarios para realizar las validaciones de usuario e iniciar la sesión de documentum,
- El parámetro **<id_documento>**es obligatorio.
- Los siguientes parámetros son opcionales:
  - Lista de parámetros **<permisos>** que contendrá los usuarios o grupos a los que se le van a dar permisos de diferente tipo en el documento especificado, indicando los valores del **usuario_documento** de la siguiente forma:
    - **permiso**: Indicaremos las competencias máximas que el usuario tendrá en el documento
    - **nombre:** Nombre del usuario o grupo al que se darán permisos en el documento,
    - **permiso_extendido:** Se indicará el permisos extendido que el usuario tendrá sobre el documento
- Lista de parámetros de **<restricciones>** que contendra los usuarios o grupos a los que se restringirá los permisos en el documento.

- **Casos de Prueba**

Caso 1: Modificación de los permisos de un documento

```
<?xml version="1.0" encoding="UTF-8"?>
        <content xsi:noNamespaceSchemaLocation="permisosdocumento.xsd"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
                <request>
                        <datos_validacion>
                                <usuario_valid>dmadmin</usuario_valid>
                                <psw_validacion>dmadmin</psw_validacion>
                                <docbase>ICM</docbase>
                        </datos_validacion>
                        <id_documento>090017728000e538</id_documento>
                        <permisos>
                                <usuario_documento permiso="Delete" nombre="dmadmin"
        permiso_extendido = “DELETE_OBJECT”/>
                        </permisos>
                           <restricciones>
                           </restricciones>

                </request>
        </content>
```

4.2.13 USO DEL SERVICIO WEB DE GESTION DE CARPETAS

4.2.13.1 DESCRIPCION DEL METODO

Permite añadir, modificar y borrar cabinet y fólder

String gestionCarpetas(String strXml)

- **Parámetros de entrada:**

Parámetro	Descripción	Tipo	Obligatorio
strXml	Xml de entrada, según schema **GestionCarpetas.xsd**	String	SI

- **Retorno:**

String - XML con la respuesta de la operación, indicando si la operacion ha sido correcta o no. Contiene dos parámetros: resultado y mensaje. Si el valor del parámetro resultado es 0, indica que no ha habido ningún problema, en caso de tener otro valor, indica que se ha producido un error, descrito en el valor del parámetro mensaje. Este xml sigue el schema **GestionCarpetas.xsd**

XML DE RETORNO

```
<?xml version="1.0" encoding="UTF-8"?>
<content>
    <response>
        <resultado>0</resultado>
        <mensaje>String</mensaje>
    </response>
</content>
```

- **Schema**

Schema GestionCarpetas.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XMLSPY v2004 rel. 3 U (http://www.xmlspy.com) by Fernando Humanes Pereira
(Comunidad de Madrid) -->
<!--Sample XML file generated by XMLSPY v2004 rel. 3 U (http://www.xmlspy.com)-->
<content xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="D:\ENTREGAS\INDRA\DOCU_20080922\docu_util_lib\fuentes\xsd\
GestionCarpetas.xsd">
    <request sesion="String" cerrarSesion="1">
        <datosValidacion>
            <usuarioValid>String</usuarioValid>
            <pswValid>String</pswValid>
            <docBase>String</docBase>
        </datosValidacion>
        <idObj>String</idObj>
        <tipoObj>String</tipoObj>
        <nombreObj>String</nombreObj>
        <pathPadre>String</pathPadre>
        <aclDomain>String</aclDomain>
        <aclName>String</aclName>
        <operacion>String</operacion>
        <atributos>
            <atributo nombre="String" operacion="String" valorRepeating="String"
indiceRepeating="String">String</atributo>
        </atributos>

        <borrarContenido>String</borrarContenido>
        <pathOriginal>String</pathOriginal>
    </request>
</content>
```

Indicaciones:

- Los parámetros ***<usuario_valid>, <psw_validacion> y <docbase>*** son necesarios para realizar las validaciones de usuario e iniciar la sesión de documentum,
- El parámetro ***<operacion>*** es obligatorio, a través del cual indicamos la operación que queremos realizar, pudiendo tomar los valores de Alta, Baja o Modificación es obligatorio
- Los siguientes parámetros son opcionales:
  - ***<idObj>***, para indicar el identificador del documento sobre el cual se realizará una operación de baja o modificación.
  - ***<tipoObj>***, para indicar si se creará un cabinet o una fólder. Es importante recordar que los cabinet (dm_cabinet) extienden de las folders (dm_folder).
  - ***<nombreObj>***, indica el nombre del objeto que se desea modificar.
  - ***<pathPadre>***, se utiliza en el alta para indicar donde situar la carpeta. En el caso de las cabinet no es necesario.
  - ***<aclDomain>***, para indicar el dominio de la ACL, que se asignara al objeto a crear o modificar.
  - ***<aclName>***, para indicar el ACL que se asignará al objeto a crear o modificar.
  - ***<borrarContenido>***, se utiliza para forzar el borrado de una fólder o cabinet independientemente de si tiene o no contenido.
  - ***< pathOriginal>***, para indicar el path original de una carpeta en las operaciones en las que quieres cambiar una carpeta de una ubicación a otra.
  - Lista ***<atributos>***, para modificar los valores de los campos que se desee en las tablas dm_cabinet o dm_folder. Para ello solo se escribe en la etiqueta ***atributo***, en el campo ***nombre***, el nombre del campo de la tabla y asignarle el valor deseado, el campo ***operación*** sólo permite los valores **ADD, MODIFY o REMOVE**, los campos ***valorRepeating*** e ***indiceRepeating*** son excluyentes, el servicio comprueba siempre el segundo y en caso de estar informado no evalúa el primero.

- **Casos de Prueba**

Caso 1: Crear un cabinet.

```
<?xml version="1.0" encoding="UTF-8"?>
<content xsi:noNamespaceSchemaLocation="GestionCarpetas.xsd"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
        <request>
                <datosValidacion>
                        <usuarioValid>dmadmin</usuarioValid>
                        <pswValid>dmadmin</pswValid>
                        <docBase>ICM</docBase>
                </datosValidacion>
                <tipoObj>dm_cabinet</tipoObj>
                <nombreObj>CABINETejg</nombreObj>
                <aclDomain>dm_dbo</aclDomain>
                <aclName>dm_acl_superusers</aclName>
                <operacion>alta</operacion>
                <atributos>
                        <atributo nombre="subject">ejg</atributo>
                        <atributo nombre="title">cabinet uf</atributo>
                </atributos>
        </request>
</content>
```

Caso 2: Modificar una fólder

```
<?xml version="1.0" encoding="UTF-8"?>
<content xsi:noNamespaceSchemaLocation="GestionCarpetas.xsd"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
        <request>
                <datosValidacion>
                        <usuarioValid>dmadmin</usuarioValid>
                        <pswValid>dmadmin</pswValid>
                        <docBase>ICM</docBase>
                </datosValidacion>
                <idObj>0b0017728000ed49</idObj>
                <tipoObj>dm_folder</tipoObj>
                <nombreObj>Folder300407_2_modif</nombreObj>
                <pathPadre>/CABINETejg</pathPadre>
                <aclName>Global User Default ACL</aclName>
                <operacion>modificacion</operacion>
                <atributos>
                        <atributo nombre="subject">ejg 2_modificado</atributo>
                        <atributo nombre="title">titulo_modificado</atributo>
<atributo nombre="atr_repeating" operación=”MODIFY” valorRepeating=”valor1” >valorNuevo
</atributo>
                </atributos>
        </request>
</content>
```

Caso 3: Eliminar un cabinet.

```
<?xml version="1.0" encoding="UTF-8"?>
<content xsi:noNamespaceSchemaLocation="GestionCarpetas.xsd"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
        <request>
                <datosValidacion>
                        <usuarioValid>dmadmin</usuarioValid>
                        <pswValid>dmadmin</pswValid>
                        <docBase>ICM</docBase>
                </datosValidacion>
                <idObj>0c00177280004bc9</idObj>
                <operacion>baja</operacion>
        </request>
</content>
```

4.2.14 USO DEL SERVICIO WEB DE VER RENDITION

Este Servicio permite consultar las renderizaciones de un documento dado su id de rendicion. La forma de llamar a este Web Service es llamar al servicio VerRendition en su operación verRendicion, a la cual se le pasará el identificador de la rendición como parámetros en un xml. El retorno de este WebService es un objeto en el cual en su primera posición está el Datahandler del contenido de la renderización consultada, que el cliente se encargará

de transformar en un fichero físico para guardar en local y visualizarlo, y en la segunda posición devuelve un xml con el resultado de la operación realizada. La forma de llamar a este Web Service es llamar al servicio VerRendition en su operación **verRendicion.**

4.2.14.1 DESCRIPCION DEL METODO

Permite obtener el contenido de una rendición

Object[] verRendicion(String strXml)

- **Parámetros de entrada:**

Parámetro	Descripción	Tipo	Obligatorio
strXml	Xml de entrada, según schema **VerRendition.xsd**	String	SI

- **Retorno:**

Object[ ] – objeto, en el que en su primera posición se encuentra el DataHandler del documento que contiene la renderización y que el cliente transformará en un fichero físico que guardará en local y visualizará, la segunda posición del objeto estará informada con un XML que contiene la respuesta de la operación, indicando si la operacion ha sido correcta o no. Contiene tres parámetros: resultado, mensaje y formato. Si el valor del parámetro resultado es 0, indica que no ha habido ningún problema, en caso de tener otro valor, indica que se ha producido un error, descrito en el valor del parámetro mensaje. El parámetro *formato* contiene el tipo de formato de la renderización del documento. Este xml sigue el schema **VerDocumento.xsd**

XML DE RETORNO

```
<?xml version="1.0" encoding="UTF-8"?>
<content>
    <response>
        <resultado>0</resultado>
        <mensaje>String</mensaje>
        <formato>pdf</formato>
    </response>
</content>
```

- **Schema VerRendition.xsd**

Schema VerDocumento.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
attributeFormDefault="unqualified">
    <xs:element name="request">
        <xs:complexType>
            <xs:all>
                <xs:element ref="datos_validacion"/>
                <xs:element ref="id_rendicion" type="xs:string"/>
                </xs:all>
            <xs:attribute name="session" type="xs:string"/>
            <xs:attribute name="cerrar_sesion" type="xs:boolean"/>
        </xs:complexType>
    </xs:element>
    <xs:element name="datos_validacion">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="usuario_valid" type="xs:string"/>
                <xs:element name="psw_validacion" type="xs:string"/>
                <xs:element name="docbase" type="xs:string"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="response">
        <xs:complexType>
            <xs:all>
                <xs:element ref="resultado" type="xs:string"/>
                <xs:element ref="mensaje" type="xs:string"/>
                <xs:element ref="formato" type="xs:string" miniOccurs=”0” />
            </xs:all>
            <xs:attribute name="cerrar_sesion" type="xs:boolean"/>
            <xs:attribute name="session" type="xs:string"/>
        </xs:complexType>
    </xs:element>
    <xs:element name="content">
        <xs:complexType>
            <xs:choice>
                <xs:element ref="request"/>
                <xs:element ref="response"/>
            </xs:choice>
        </xs:complexType>
    </xs:element>
    <xs:element name="id_rendicion" type="xs:string"/>
    <xs:element name="resultado" type="xs:string"/>
    <xs:element name="mensaje" type="xs:string"/>
    <xs:element name="formato" type="xs:string"/>
</xs:schema>
```

Indicaciones:

- Los parámetros ***<usuario_valid>, <psw_validacion> y <docbase>*** son necesarios para realizar las validaciones de usuario e iniciar la sesión de documentum,
- El parámetro **<id_rendicion>** es obligatorio.

- **Casos de Prueba**

Caso 1 –Visualización de Rendición

```
<?xml version="1.0" encoding="UTF-8"?>
<content xsi:noNamespaceSchemaLocation="verRenditionsara.xsd" xmlns:xsi="http://ww
w.w3.org/2001/XMLSchema-instance">
    <request>
        <datos_validacion>
            <usuario_valid>dmadmin</usuario_valid>
            <psw_validacion>dmadmin</psw_validacion>
            <docbase>ICM</docbase>
        </datos_validacion>
        <id_rendicion>0900177280004ba2</id_rendicion>
    </request>
</content>
```

4.3 PAQUETES COMUNES

En este apartado se describen las clases java comunes utilizados por los distintos Servicios Web.

4.3.1 Paquete docu_ws.ws.icm.config

Clase	Descripción
LectorFormatos.java	Xml de entrada, según schema **GestionCarpetas.xsd**
EvitarSqlInyectado.java	Se lleva a cabo la validación de que no existan parámetros de los xml de entrada que puedan llevar sql embebido. Lo que se hace es que en dichos parámetros no existan palabras clave que están presentes en una Query, como son “select ”,” where ”,”delete”, “update”, etc...
LectorConfiguracion.java	Se encarga de leer el archivo de configuración del sistema en cuanto a rutas del servidor y activación de trazas
DefinicionFormatos.xml	Fichero xml donde se definen los formatos permitidos para los documentos

4.3.2 Paquete docu_ws.ws.icm.util.constants

Clase	Descripción
ErrorMessageConstants.java	Clase que contiene las constantes que van a definir el código de error y el mensaje que se asigna a una excepción. Esta formado por constantes de tipo String [] en el que el valor de la posición 0 es el código que se le asigna a la excepción y en la posición 1 se encuentra la descripción que la queremos dar
XMLParametersNameContanst.java	Clase de constantes que contiene literales con los nombres de los parámetros u otros atributos que se utilizan para leer algunos xml.

4.3.3 Paquete docu_ws.ws.icm.util.exception

Clase	Descripción
WSException.java	Clases que extienden de la clase Exception. Tienen dos atributos: *mensaje y código*

Métodos clase WSException

```
/*constructores de la clase WSException*/
public WSException(String strMsj,String strCod);
/*constructores de WSException que además guarda la excepción a partir de la que hemos lanzado la actual*/
public WSException(String strMsj,String strCod,Exception eException);

/*devuelve el código asignado a la excepción*/
public String getCodigo();

/*modifica el código asignado a la excepción*/
public void setCodigo(String strCodigo;

/*devuelve el mensaje asignado a la excepción*/
public String getMensaje();

/*modifica el mensaje asignado a la excepción*/
public void setMensaje(String strMensaje);
```

Todas las excepciones lanzadas por los servicios serán recogidas y lanzadas en forma de ***WSException*** si se producen en el servicio de búsqueda.

4.3.4 Paquete docu_ws.ws.icm.util.WebServices

Clase	Descripción
Login.java	Clase que se ocupa de la conexión y desconexión con el repositorio documental. Los métodos de esta clase se invocan en todos los servicios web. La sesión contra el repositorio documental se crea y se destruye dentro de cada servicio.

4.3.5 Paquete docu_ws.ws.icm.util.logs

Clase	Descripción
Trazas.java	Crea una instancia del log4j para las trazas de la aplicación

4.3.6 Paquete docu_ws.ws.icm.xml

Conjunto de clases y paquetes autogenerados a partir de la herramienta JAXB, utilizados para el manejo del fichero **DefinicionFormatos.xml** situado en **docu_ws.ws.icm.config**. Todas estas clases y paquetes son de uso interno de la librería

5 SOLUCION DE INTEGRACION DOCU_LIB

5.1 INSTALACION Y CONFIGURACION

5.1.1 LIBRERÍAS NECESARIAS

Las librerías necesarias para el correcto funcionamiento de las aplicaciones j2ee según Framework 2 que utilizan la librería docu_lib proporcionada por icm para acceder a Documentum son las descritas en la siguiente tabla:

Nombre	Motivo	Localización en SOJA	C	E
docu_util_lib.jar	Para generar los xml que se le pasan como parámetro a los distintos servicios	https://gestiona.madrid.org/soja_int/run/j/EnlaceRecurso.icm?cd_recurso=1542	X	X
jaxb_icm_1_3.jar	Manejo de xml	https://gestiona.madrid.org/soja_int/run/j/EnlaceLibreria.icm?cd_libreria=181	X	X
log4j-1.2.15.jar	Paquete de librerías Axis	Todas estas librerías se han empaquetado en lib_dep_docu_lib.zip y se han colgado en https://gestiona.madrid.org/soja_int/html/web/EnlaceLibreria.icm?cd_elemento=741		X
activation.jar	Relacionada con el servicio de Importación de documentos (Datahandler)		X	X
mail-1.4.jar	Relacionada con el servicio de Importación de documentos (Datahandler)		X	X
dfc.jar	Perteneciente al api de Documentum		X	X
ctsTransform.jar	Perteneciente al api de Documentum		X	X
certjFIPS.jar	Dependencia del api de Documentum			X
jsafeFIPS.jar	Dependencia del api de Documentum			X
aspectjrt.jar	Dependencia del api de Documentum			X

C: Librerías utilizadas en Compilación.
E: Librerías utilizadas en Ejecución.
Nota: Además de estas librerías se podrán utilizar cualquier librería autorizada para el Framework 2

5.1.2 INSTALACION

La librería **docu_lib.jar** puede descargársela de:
https://gestiona.madrid.org/soja_int/run/j/EnlaceRecurso.icm?cd_recurso=1182

5.1.3 CONFIGURACION

Las variables que pueden venir en el fichero de configuración de la aplicación son las indicadas en la siguiente tabla.

Variables	Descripción	Valores que toma	Obligatorio
documentum.docbase	Nombre del repositorio	desdocum_01	SI
documentum.usuario	Normalmente la autenticación en documentum se realizará a través de LDAP. Esta variable sólo vendrá en el caso de necesitar un usuario genérico de conexión con Documentum.	xxxx_<usuario> siendo xxxx el nombre del proyecto	NO
documentum.clave	Clave del usurio encriptada del usuario genérico.		NO
NOTA: Todas las variables se declararán en el apartado /*************************/ /* DOCUMENTUM */ /*************************/. Cualquier otra variable que se necesite definir seguirá la siguiente nomenclatura **documentum.<descripción_variable>**			

Para la correcta ejecución de los proyectos que utilicen esta solución (docu_lib), se debe incluir el fichero **dfc.properties** en el directorio classes de la aplicación que la utilice. A continuación se muestra un ejemplo de fichero dfc.properties en el que se muestran las variables que como mínimo deben informarse.

dfc.properties

```
dfc.docbroker.host[0]=icmcs03
dfc.docbroker.port[0]=1489
dfc.globalregistry.repository=desdocum_01
dfc.globalregistry.username=dm_bof_registry
dfc.globalregistry.password=GrRNPhLJrkoTDAZE0RGJow\=\=
```

5.2 USO

5.2.1 GENERAL

Para llamar a cada uno de los servicios se hace mediante la clase y método de cada uno de ellos, pasándoles los siguientes parámetros de entrada:

- **String con el xml que contiene los datos de la operación**,
- **Logger:** Instancia de la clase Logger para la escritura de trazas y mensajes de error

Cada uno de los métodos, previamente a realizar la operación que le corresponda, realiza la conexión con el repositorio, y posteriormente se desconecta del mismo, devolviendo un string que contiene un xml con el código de resultado de la operación y una descripción.

5.2.2 GENERACIÓN DE FICHEROS XML DE ENTRADA A LOS SERVICIOS

Para la creación y manejo de los ficheros xml que se le pasarán como parámetro a cada uno de los servicios se proporciona la librería **docu_util_lib.jar** que mediante el uso de JAXB realiza las operaciones necesarias para el Manejo de información mediante XML
Está publicada en https://gestiona.madrid.org/soja_int/html/web/EnlaceLibreria.icm?cd_elemento=723
Para generar estos XML se tienen que seguir las especificaciones propuestas por ICM y que se detallan en el

Manual_manejo_XML.pdf publicado en
https://gestiona.madrid.org/soja_int/run/j/EnlaceManual.icm?cd_elemento=781
Estos xml además deben cumplir los esquemas de los mismos publicados como ***Esquemas del FrameWork de servicios Documentales*** en:
https://gestiona.madrid.org/soja_int/html/web/EnlaceRecurso.icm?cd_recurso=1142

5.1.1 USO DEL SERVICIO DE ADMINISTRACION DE GRUPOS

Este Servicio permite crear y modificar grupos. La forma de llamar a este Servicio es llamar al método **administrarGrupo** de la clase **administración** pasándole como parámetros, un string con el xml que contiene los datos necesarios para la ejecución y un objeto logger para escribir los errores.

5.1.1.1 DESCRIPCION DEL METODO

Permite crear y modificar grupos

String administrarGrupo(String strXml, Logger log)

Un ejemplo de llamada al método **administrarGrupo** se encuentra en el siguiente código java. En este se indica la ruta temporal donde se encuentra el fichero xml, este fichero se pasa a un String que será un parámetro de entrada del método

Ejemplo de llamada al método administarGrupo

El fichero xml (toda su ruta) es previamente transformado en string como:

```
String ruta = Config.get("aplicacion.download")+"administracionGrupos.xml";
FileInputStream fileinputstream = new FileInputStream(fichruta);
byte abyte0[] = new byte[fileinputstream.available()];
fileinputstream.read(abyte0);
fileinputstream.close();
s = new String(abyte0);
s = s.trim();

Object[] obj = administracion.administrarGrupo(s, logger);
```

- **Parámetros de entrada:**

Parámetro	Descripción	Tipo	Obligatorio
strXml	Xml de entrada, según schema **AdministracionGrupos.xsd**	String	SI
log	Objeto Logger para escribir los errores	Logger	SI

- **Retorno:**

String - XML con la respuesta de la operación, indicando si la operación ha sido correcta o no. Contiene dos parámetros: resultado y mensaje. Si el valor del parámetro resultado es 0, indica que no ha habido ningún problema, en caso de tener otro valor, indica que se ha producido un error, descrito en el valor del parámetro mensaje. Este xml sigue el schema **AdministracionGrupos.xsd**

XML DE RETORNO

```
<?xml version="1.0" encoding="UTF-8"?>
<content>
    <response>
        <resultado>0</resultado>
        <mensaje>String</mensaje>
    </response>
</content>
```

- **Schema AdministracionGrupos.xsd**

Schema AdministracionGrupos.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XMLSpy v2007 (http://www.altova.com) by UF787648 (EMBRACE) -->
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
attributeFormDefault="unqualified">
    <xs:element name="datos_validacion">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="usuario_valid" type="xs:string"/>
                <xs:element name="psw_validacion" type="xs:string"/>
                <xs:element name="docbase" type="xs:string"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="response">
        <xs:complexType>
            <xs:all>
                <xs:element ref="resultado"/>
                <xs:element ref="mensaje"/>
            </xs:all>
            <xs:attribute name="cerrar_sesion" type="xs:boolean"/>
            <xs:attribute name="sesion" type="xs:string"/>
        </xs:complexType>
    </xs:element>
    <xs:element name="grupo">
        <xs:complexType>
            <xs:attribute name="nombre" type="xs:string"/>
            <xs:attribute name="accion" type="xs:string"/>
        </xs:complexType>
    </xs:element>
    <xs:element name="lista_grupos">
        <xs:complexType>
            <xs:sequence>
                <xs:element ref="grupo" minOccurs="0" maxOccurs="unbounded"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
```

```
<xs:element name="usuario">
        <xs:complexType>
            <xs:attribute name="nombre" type="xs:string"/>
            <xs:attribute name="accion" type="xs:string"/>
        </xs:complexType>
</xs:element>
<xs:element name="lista_usuarios">
        <xs:complexType>
            <xs:sequence>
                <xs:element ref="usuario" minOccurs="0" maxOccurs="unbounded"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="content">
        <xs:complexType>
            <xs:choice>
                <xs:element ref="request"/>
                <xs:element ref="response"/>
            </xs:choice>
        </xs:complexType>
    </xs:element>
<xs:element name="request">
        <xs:complexType>
            <xs:all>
                <xs:element ref="id" minOccurs="0"/>
                <xs:element ref="email" minOccurs="0"/>
                <xs:element ref="descripcion"/>
                <xs:element ref="nombre_grupo" minOccurs="0"/>
                <xs:element ref="datos_validacion"/>
                <xs:element ref="lista_usuarios" minOccurs="0"/>
                <xs:element ref="lista_grupos" minOccurs="0"/>
                <xs:element ref="codigo_aplicacion" minOccurs="0"/>
                <xs:element ref="tipo_documental" minOccurs="0"/>
            </xs:all>
            <xs:attribute name="session" type="xs:string"/>
            <xs:attribute name="cerrar_sesion" type="xs:boolean"/>
        </xs:complexType>
</xs:element>
<xs:element name="id" type="xs:string"/>
<xs:element name="email" type="xs:string"/>
<xs:element name="descripcion" type="xs:string"/>
<xs:element name="nombre_grupo" type="xs:string"/>
    <xs:element name="codigo_aplicacion" type="xs:string"/>
    <xs:element name="tipo_documental" type="xs:string"/>
    <xs:element name="resultado" type="xs:int"/>
    <xs:element name="mensaje" type="xs:string"/>
</xs:schema>
```

Indicaciones:

- Los parámetros ***<usuario_valid>, <psw_validacion> y <docbase>*** son necesarios para realizar las validaciones de usuario e iniciar la sesión de documentum
- El parámetro ***<descripcion>*** es obligatorio

- Los parámetros **<id>** y **<nombre_grupo>** son excluyentes.
- Si se desea dar de alta un grupo se utilizatá el parámetro **<nombre_grupo>** donde se informará del nombre del grupo que se quiere crear.
- Para modificar un grupo se incluirá el parámetro **<id>** con el identificador del grupo a modificar y se omitirá **<nombre_grupo>**.

- **Casos de Prueba**

Caso 1: Alta de grupos

```
<?xml version="1.0" encoding="UTF-8"?>
  <content xsi:noNamespaceSchemaLocation="AdministracionGrupos.xsd"
     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
   <request>
       <email>mtangel@indra.es</email>
       <descripcion>Prueba ws de febrero</descripcion>
    <nombre_grupo>icm_gr_frameEner</nombre_grupo>
        <datos_validacion>
              <usuario_valid>dmadmin</usuario_valid>
              <psw_validacion>dmadmin</psw_validacion>
              <docbase>icm</docbase>
        </datos_validacion>
     <lista_usuarios>
        <usuario accion="add" nombre="icm"/>
     </lista_usuarios>
         <lista_grupos>
        <grupo accion="add" nombre="icm_gr_config"/>
      </lista_grupos>
  </request>
</content>
```

Caso 2: Modificación de un grupo

```
<?xml version="1.0" encoding="UTF-8"?>
        <content xsi:noNamespaceSchemaLocation="AdministracionGrupos.xsd"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
        <request>
            <email>mtangel@indra.es</email>
            <descripcion>Prueba ws de febrero</descripcion>
             <id>1203a56380000900</id>
             <datos_validacion>
                    <usuario_valid>dmadmin</usuario_valid>
                    <psw_validacion>dmadmin</psw_validacion>
                    <docbase>icm</docbase>
              </datos_validacion>
              <lista_usuarios>
                      <usuario accion="add" nombre="icm"/>
              </lista_usuarios>
              <lista_grupos>
                        <grupo accion="add" nombre="icm_gr_config"/>
               </lista_grupos>
         </request>
       </content>
```

5.1.2 USO DEL SERVICIO DE BORRAR DOCUMENTO

Este Servicio permite el borrado del documento que se corresponde con el identificador del documento que se le pasa como parámetro. La forma de llamar a este Servicio es llamar al método **borrarDocumento** de la clase **borrardocumento**, pasándole como parámetros, un string con el xml que contiene los datos necesarios para la ejecución y un objeto logger para escribir los errores.

5.1.2.1 DESCRIPCION DEL METODO

Permite borrar documentos

String borrarDocumento(String strXml, Logger log)

Un ejemplo de llamada al método ***borrarDocumento*** se encuentra en el siguiente código java. En este se indica la ruta temporal donde se encuentra el fichero xml, este fichero se pasa a un String que será un parámetro de entrada del método

Ejemplo de llamada al método borrarDocumento

El fichero xml (toda su ruta) es previamente transformado en string como:

```
String ruta = Config.get("aplicacion.download")+"borrarDocumento.xml";
FileInputStream fileinputstream = new FileInputStream(fichruta);
byte abyte0[] = new byte[fileinputstream.available()];
fileinputstream.read(abyte0);
fileinputstream.close();
s = new String(abyte0);
s = s.trim();

Object[] obj = borrardocumento.borrarDocumento(s, logger);
```

- **Parámetros de entrada:**

Parámetro	Descripción	Tipo	Obligatorio
strXml	Xml de entrada, según schema **BorrarDocumento.xsd**, con los datos del documento a borrar y del usuario que lleva a cabo la operacion	String	SI
log	Objeto Logger para escribir los errores	Logger	SI

- **Retorno:**

String - XML con la respuesta de la operación, indicando si la operación ha sido correcta o no. Contiene dos parámetros: resultado y mensaje. Si el valor del parámetro resultado es 0, indica que no ha habido ningún problema, en caso de tener otro valor, indica que se ha producido un error, descrito en el valor del parámetro mensaje. Este xml sigue el schema **BorrarDocumento.xsd**.

XML DE RETORNO

```
<?xml version="1.0" encoding="UTF-8"?>
<content>
    <response>
        <resultado>0</resultado>
        <mensaje>String</mensaje>
    </response>
</content>
```

- **Schema BorrarDocumento.xsd**

Schema BorrarDocumento.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XMLSpy v2007 (http://www.altova.com) by UF787648 (EMBRACE) -->
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
attributeFormDefault="unqualified">
    <xs:element name="request">
        <xs:complexType>
            <xs:all>
                <xs:element ref="datos_validacion"/>
                <xs:element ref="id_documento"/>
                <xs:element ref="version"/>
            </xs:all>
            <xs:attribute name="session" type="xs:string"/>
            <xs:attribute name="cerrar_sesion" type="xs:boolean"/>
        </xs:complexType>
    </xs:element>
    <xs:element name="datos_validacion">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="usuario_valid" type="xs:string"/>
                <xs:element name="psw_validacion" type="xs:string"/>
                <xs:element name="docbase" type="xs:string"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="response">
        <xs:complexType>
            <xs:all>
                <xs:element ref="resultado"/>
                <xs:element ref="mensaje"/>
                <xs:element ref="nombre_documento" minOccurs="0"/>
            </xs:all>
            <xs:attribute name="cerrar_sesion" type="xs:boolean"/>
            <xs:attribute name="session" type="xs:string"/>
        </xs:complexType>
    </xs:element>
```

```
  <xs:element name="content">
        <xs:complexType>
            <xs:choice>
                <xs:element ref="request"/>
                <xs:element ref="response"/>
            </xs:choice>
        </xs:complexType>
    </xs:element>
    <xs:element name="id_documento" type="xs:string"/>
    <xs:element name="resultado" type="xs:string"/>
    <xs:element name="mensaje" type="xs:string"/>
    <xs:element name="nombre_documento"/>
    <xs:element name="version" type="xs:string"/>
</xs:schema>
```

Indicaciones:

- Los parámetros ***<usuario_valid>, <psw_validacion> y <docbase>*** son necesarios para realizar las validaciones de usuario e iniciar la sesión de documentum,
- Los parámetros **<version>** y **<id_documento>** son obligatorios
- En el parámetro **<version>**, se indicará si se quieren borrar todas las versiones de un documento o no. Su valor por defecto es **CURRENT,** con este valor se borrará el documento indicado en el parámetro **<id_documento>**
- Para borrar todas las versiones se indicará con la palabra **ALL**
- Para borrar una versión especifica se indica con el identificador del documento, y la palabra **CURRENT** en el parámetro versión.

- **Casos de Prueba**

Caso 1: Borrado de la versión concreta de un documento. Versión igual a CURRENT

```
<?xml version="1.0" encoding="UTF-8"?>
<content xsi:noNamespaceSchemaLocation=" BorrarDocumento.xsd " xmlns:xsi="http://ww
w.w3.org/2001/XMLSchema-instance">
      <request>
          <version>current</version>
           <datos_validacion>
                <usuario_valid>dmadmin</usuario_valid>
                <psw_validacion>dmadmin</psw_validacion>
                <docbase>ICM</docbase>
           </datos_validacion>
           <id_documento>0900177280004ba2</id_documento>
           </request>
</content>
```

Caso 2: Borrado de todas las versiones de un documento. Versión igual a ALL

```
<?xml version="1.0" encoding="UTF-8"?>
<content xsi:noNamespaceSchemaLocation=" BorrarDocumento.xsd " xmlns:xsi="http://ww
w.w3.org/2001/XMLSchema-instance">
    <request>
        <version>all</version>
        <datos_validacion>
            <usuario_valid>dmadmin</usuario_valid>
            <psw_validacion>dmadmin</psw_validacion>
            <docbase>ICM</docbase>
        </datos_validacion>
        <id_documento>0900177280004ba2</id_documento>
    </request>
</content>
```

5.1.3 USO DEL SERVICIO DE PEDIR RENDITION

Este Servicio permite generar transformaciones de documentos a formatos .pdf y .html. La forma de llamar a este Web Service es llamar al servicio PedirRendition en su operación **pedirRendition**, a la cual se le pasará el identificador del documento a transformar y el formato deseado como parámetros en un xml

5.1.3.1 DESCRIPCION DEL METODO

Permite generar transformaciones de documentos a formatos .pdf y .html

String pedirRendition(String strXml, Logger log) throws WSException

Un ejemplo de llamada al método ***pedirRendition*** se encuentra en el siguiente código java. En este se indica la ruta temporal donde se encuentra el fichero xml, este fichero se pasa a un String que será un parámetro de entrada del método.

Ejemplo de llamada al método pedirRendition

El fichero xml (toda su ruta) es previamente transformado en string como:

```
 String ruta = Config.get("aplicacion.download")+" pedirRendition.xml";
FileInputStream fileinputstream = new FileInputStream(fichruta);
byte abyte0[] = new byte[fileinputstream.available()];
fileinputstream.read(abyte0);
fileinputstream.close();
s = new String(abyte0);
s = s.trim();

Object[] obj = rendition.pedirRendition(s,logger );
```

- **Parámetros de entrada:**

Parámetro	Descripción	Tipo	Obligatorio
strXml	Xml de entrada, según schema **PedirRendition.xsd**	String	SI
Log	Objeto Logger para escribir los errores	Logger	SI

- **Retorno:**

String - XML con la respuesta de la operación, indicando si la operación ha sido correcta o no. Contiene dos parámetros: resultado y mensaje. Si el valor del parámetro resultado es 0, indica que no ha habido ningún problema, en caso de tener otro valor, indica que se ha producido un error, descrito en el valor del parámetro mensaje. Este xml sigue el schema **PedirRendition.xsd**

XML DE RETORNO

```
<?xml version="1.0" encoding="UTF-8"?>
<content>
    <response>
        <resultado>0</resultado>
        <mensaje>String</mensaje>
    </response>
</content>
```

- **Schema PedirRendition.xsd**

Schema PedirRendition.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XMLSpy v2007 (http://www.altova.com) by UF787648 (EMBRACE) -->
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
attributeFormDefault="unqualified">
    <xs:element name="request">
        <xs:complexType>
            <xs:all>
                <xs:element ref="datos_validacion"/>
                <xs:element ref="id_documento"/>
                <xs:element ref="formato"/>
            </xs:all>
            <xs:attribute name="session" type="xs:string"/>
            <xs:attribute name="cerrar_sesion" type="xs:boolean"/>
        </xs:complexType>
    </xs:element>
```

```
<xs:element name="datos_validacion">
      <xs:complexType>
         <xs:sequence>
            <xs:element name="usuario_valid" type="xs:string"/>
            <xs:element name="psw_validacion" type="xs:string"/>
            <xs:element name="docbase" type="xs:string"/>
         </xs:sequence>
      </xs:complexType>
   </xs:element>
  <xs:element name="response">
      <xs:complexType>
         <xs:all>
            <xs:element ref="resultado"/>
            <xs:element ref="mensaje"/>
         </xs:all>
         <xs:attribute name="cerrar_sesion" type="xs:boolean"/>
         <xs:attribute name="session" type="xs:string"/>
      </xs:complexType>
   </xs:element>
   <xs:element name="datos_validacion">
      <xs:complexType>
         <xs:sequence>
            <xs:element name="usuario_valid" type="xs:string"/>
            <xs:element name="psw_validacion" type="xs:string"/>
            <xs:element name="docbase" type="xs:string"/>
         </xs:sequence>
      </xs:complexType>
   </xs:element>
   <xs:element name="response">
      <xs:complexType>
         <xs:all>
            <xs:element ref="resultado"/>
            <xs:element ref="mensaje"/>
         </xs:all>
         <xs:attribute name="cerrar_sesion" type="xs:boolean"/>
         <xs:attribute name="session" type="xs:string"/>
      </xs:complexType>
   </xs:element>
   <xs:element name="content">
      <xs:complexType>
         <xs:choice>
            <xs:element ref="request"/>
            <xs:element ref="response"/>
         </xs:choice>
      </xs:complexType>
   </xs:element>
    <xs:element name="id_documento" type="xs:string"/>
   <xs:element name="resultado" type="xs:string"/>
   <xs:element name="mensaje" type="xs:string"/>
   <xs:element name="formato" type="xs:string"/>
</xs:schema>
```

Indicaciones:

- Los parámetros ***<usuario_valid>, <psw_validacion> y <docbase>*** son necesarios para realizar las

validaciones de usuario e iniciar la sesión de documentum,

- El parámetro <id_documento> es obligatorio
- Los posibles valores para el parámetro <formato> son PDF o HTML

- **Casos de Prueba**

Caso 1: Petición de rendition. Todos los datos correctos. Formato pdf

```
<?xml version="1.0" encoding="UTF-8"?>
<content xsi:noNamespaceSchemaLocation="pedirRendition.xsd" xmlns:xsi="http://ww
w.w3.org/2001/XMLSchema-instance">
    <request>
        <datos_validacion>
            <usuario_valid>dmadmin</usuario_valid>
            <psw_validacion>dmadmin</psw_validacion>
            <docbase>ICM</docbase>
        </datos_validacion>
        <formato>pdf</formato>
        <id_documento>09001772800053b7</id_documento>
    </request>
</content>
```

Caso 2: Petición de rendition. Todos los datos correctos. Formato html

```
<?xml version="1.0" encoding="UTF-8"?>
<content xsi:noNamespaceSchemaLocation="pedirRendition.xsd" xmlns:xsi="http://ww
w.w3.org/2001/XMLSchema-instance">
    <request>
        <datos_validacion>
            <usuario_valid>dmadmin</usuario_valid>
            <psw_validacion>dmadmin</psw_validacion>
            <usuario>dmadmin</usuario>
            <docbase>ICM</docbase>
        </datos_validacion>
        <formato>html</formato>
        <id_documento>09001772800053b7</id_documento>
    </request>
</content>
```

5.1.4 USO DEL SERVICIO DE VER DOCUMENTO

Este Servicio permite la visualización del documento que se corresponde con el identificador del documento que se le pasa como parámetro en un xml. El retorno de este Servicio es un objeto en el cual en su primera posición está el Datahandler del contenido del documento, que el cliente se encargará de transformar en un fichero físico para guardar en local y visualizarlo, y en la segunda posición devuelve un xml en el que nos indica el nombre del documento visualizado. La forma de llamar a este Service es llamar al servicio VerDocumento en su operación **verDocumento.**

5.1.4.1 DESCRIPCION DEL METODO

Permite visualizar documentos

Object[] verDocumento(String strXml, Logger log)

Un ejemplo de llamada al método ***verDocumento*** se encuentra en el siguiente código java. En este se indica la ruta temporal donde se encuentra el fichero xml, este fichero se pasa a un String que será un parámetro de entrada del método.

Ejemplo de llamada al método verDocumento

El fichero xml (toda su ruta) es previamente transformado en string como:

```
String ruta = Config.get("aplicacion.download")+" verDocumento.xml";
FileInputStream fileinputstream = new FileInputStream(fichruta);
byte abyte0[] = new byte[fileinputstream.available()];
fileinputstream.read(abyte0);
fileinputstream.close();
s = new String(abyte0);
s = s.trim();

Object[] obj = ver.verDocumento(s, logger);
```

- **Parámetros de entrada:**

Parámetro	Descripción	Tipo	Obligatorio
strXml	Xml de entrada, según schema **VerDocumento.xsd**	String	SI
log	Objeto Logger para escribir los errores	Logger	SI

- **Retorno:**

Object[ ] – objeto, en el que en su primera posición se encuentra el DataHandler del documento que se quiere visualizar y que el cliente transformará en un fichero físico que guardará en local y visualizará, la segunda posición del objeto estará informada con un XML que contiene la respuesta de la operación, indicando si la operacion ha sido correcta o no. Contiene tres parámetros: nombre_documento, resultado y mensaje. Si el valor del parámetro resultado es 0, indica que no ha habido ningún problema, en caso de tener otro valor, indica que se ha producido un error, descrito en el valor del parámetro mensaje. El parámetro *nombre_documento* contiene el nombre del documento. Este xml sigue el schema **VerDocumento.xsd**

XML DE RETORNO

```
<?xml version="1.0" encoding="UTF-8"?>
<content>
    <response>
        <resultado>0</resultado>
        <mensaje>String</mensaje>
        <nombre_documento>String</nombre_documento>
    </response>
</content>
```

- **Schema VerDocumento.xsd**

Schema VerDocumento.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XMLSpy v2007 (http://www.altova.com) by UF787648 (EMBRACE) -->
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
attributeFormDefault="unqualified">
    <xs:element name="request">
        <xs:complexType>
            <xs:all>
                <xs:element ref="datos_validacion"/>
                <xs:element name="id_documento" type="xs:string"/>
            </xs:all>
            <xs:attribute name="session" type="xs:string"/>
            <xs:attribute name="cerrar_sesion" type="xs:boolean"/>
        </xs:complexType>
    </xs:element>
    <xs:element name="datos_validacion">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="usuario_valid" type="xs:string"/>
                <xs:element name="psw_validacion" type="xs:string"/>
                <xs:element name="docbase" type="xs:string"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
     <xs:element name="response">
        <xs:complexType>
            <xs:all>
                <xs:element name="resultado" type="xs:string"/>
                <xs:element name="mensaje" type="xs:string"/>
                <xs:element name="nombre_documento" type="xs:string" minOccurs="0"/>
            </xs:all>
           <xs:attribute name="cerrar_sesion" type="xs:boolean"/>
            <xs:attribute name="session" type="xs:string"/>
        </xs:complexType>
    </xs:element>
    <xs:element name="content">
        <xs:complexType>
            <xs:choice>
                <xs:element ref="request"/>
                <xs:element ref="response"/>
            </xs:choice>
        </xs:complexType>
    </xs:element>
</xs:schema>
```

Indicaciones:

- Los parámetros ***<usuario_valid>, <psw_validacion> y <docbase>*** son necesarios para realizar las validaciones de usuario e iniciar la sesión de documentum,
- El parámetro **<id_documento>** es obligatorio.
- El resto de parámetros son opcionales.

- **Casos de Prueba**

Caso 1 –Visualización de documento.

```
<?xml version="1.0" encoding="UTF-8"?>
<content xsi:noNamespaceSchemaLocation="verdocumento.xsd" xmlns:xsi="http://ww
w.w3.org/2001/XMLSchema-instance">
    <request>
        <datos_validacion>
            <usuario_valid>dmadmin</usuario_valid>
            <psw_validacion>dmadmin</psw_validacion>
            <docbase>ICM</docbase>
        </datos_validacion>
        <id_documento>0900177280004ba2</id_documento>
    </request>
</content>
```

5.1.5 USO DEL SERVICIO DE TABLAS EXTERNAS

Este Servicio permite realizar modificaciones, inserciones y borrados en tablas externas. La forma de llamar a este Servicio es llamar al método **gestiontablas** de la clase **gestiontablas** pasándole como parámetros, un string con el xml que contiene los datos necesarios para la ejecución y un objeto logger para escribir los errores.

5.1.5.1 DESCRIPCION DEL METODO

String gestionTablas(String strXml, Logger log)

Un ejemplo de llamada al método ***gestiontablas*** se encuentra en el siguiente código java. En este se indica la ruta temporal donde se encuentra el fichero xml, este fichero se pasa a un String que será un parámetro de entrada del método.

Ejemplo de llamada al método gestionTablas

El fichero xml (toda su ruta) es previamente transformado en string como:

```
 String ruta = Config.get("aplicacion.download")+" gestionTablas.xml";
FileInputStream fileinputstream = new FileInputStream(fichruta);
byte abyte0[] = new byte[fileinputstream.available()];
fileinputstream.read(abyte0);
fileinputstream.close();
s = new String(abyte0);
s = s.trim();

Object[] obj = gestiontablas.gestiontablas(s, logger);
```

- **Parámetros de entrada:**

Parámetro	Descripción	Tipo	Obligatorio
strXml	Xml de entrada, según schema **GestionTablas.xsd**	String	SI
log	Objeto Logger para escribir los errores	Logger	SI

- **Retorno:**

String - XML con la respuesta de la operación, indicando si la operacion ha sido correcta o no. Contiene dos parámetros: resultado y mensaje. Si el valor del parámetro resultado es 0, indica que no ha habido ningún problema, en caso de tener otro valor, indica que se ha producido un error, descrito en el valor del parámetro mensaje. Este xml sigue el schema **GestionTablas.xsd**

XML DE RETORNO

```
<?xml version="1.0" encoding="UTF-8"?>
<content>
    <response>
        <resultado>0</resultado>
        <mensaje>String</mensaje>
    </response>
</content>
```

- **Schema GestionTablas.xsd**

Schema GestionTablas.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XMLSpy v2007 (http://www.altova.com) by UF787648 (EMBRACE) -->
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
attributeFormDefault="unqualified">
    <xs:element name="datos_validacion">
        <xs:complexType>
            <xs:all>
                <xs:element name="usuario_valid" type="xs:string"/>
                <xs:element name="psw_validacion" type="xs:string"/>
                <xs:element name="docbase" type="xs:string"/>
            </xs:all>
        </xs:complexType>
    </xs:element>
    <xs:element name="response">
        <xs:complexType>
            <xs:all>
                <xs:element name="resultado" type="xs:int"/>
                <xs:element name="mensaje" type="xs:string"/>
            </xs:all>
            <xs:attribute name="session" type="xs:string"/>
            <xs:attribute name="cerrar_sesion" type="xs:boolean"/>
        </xs:complexType>
    <xs:element name="lista_Busqueda">
        <xs:complexType>
            <xs:sequence>
                <xs:element ref="campo_busqueda" minOccurs="0" maxOccurs="unbounded"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="operacion" type="xs:string"/>
```

```
<xs:element name="request">
  <xs:complexType>
    <xs:all>
      <xs:element ref="datos_validacion"/>
      <xs:element name="tabla" type="xs:string"/>
      <xs:element ref="operacion"/>
      <xs:element ref="lista_Atributos" minOccurs="0"/>
      <xs:element ref="lista_Busqueda" minOccurs="0"/>
    </xs:all>
    <xs:attribute name="session" type="xs:string"/>
    <xs:attribute name="cerrar_sesion" type="xs:boolean"/>
  </xs:complexType>
</xs:element>
<xs:element name="valor_campo" type="xs:string"/>
<xs:element name="campo_busqueda">
  <xs:complexType>
    <xs:simpleContent>
      <xs:extension base="xs:string">
        <xs:attribute name="nombre" type="xs:string"/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
</xs:element>
<xs:element name="campo">
  <xs:complexType>
    <xs:simpleContent>
      <xs:extension base="xs:string">
        <xs:attribute name="nombre" type="xs:string"/>
        <xs:attribute name="clave" type="xs:boolean"/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
</xs:element>
<xs:element name="tabla">
  <xs:complexType>
    <xs:all>
      <xs:element name="tabla_relacion" type="xs:string"/>
      <xs:element name="campo_relacion" type="xs:string"/>
      <xs:element name="campo_origen" type="xs:string"/>
    </xs:all>
  </xs:complexType>
</xs:element>
<xs:element name="content">
  <xs:complexType>
    <xs:choice>
      <xs:element ref="request"/>
      <xs:element ref="response"/>
    </xs:choice>
  </xs:complexType>
</xs:element>
</xs:schema>
```

Indicaciones:

- Los parámetros ***<usuario_valid>, <psw_validacion> y <docbase>*** son necesarios para realizar las

validaciones de usuario e iniciar la sesión de documentum,

- El parámetro ***<tabla>*** es obligatorio, sirve para indicar el nombre de la tabla sobre la que se realiza la operación.
- El parámetro ***<operacion>*** es obligatorio, sirve para indicar la operación a realizar. Los valores que puede tomar son: **Insert, Delete, Update**.
- El parámetro ***<lista_Atributos>*** es opcional, esta lista se utiliza en el caso de que la operación que se realice sea una inserción o una modificación. En el caso de las modificaciones, contendrá el nuevo valor a asignar (el SET de la sentencia UPDATE). En el caso de una inserción y que la tabla tratada tenga algún tipo de código automático se debe introducir este a 0 si queremos que salte el triggers de la tabla, aunque también podemos introducir de manera manual el mismo. Para hacer dicha operaciones se rellena la etiqueta ***campo*** asignando en ***nombre*** el valor del campo de la tabla y su valor, en la etiqueta clave se especifica si ese campo es clave en la tabla o no.
- El parámetro ***<lista_Busqueda>*** es opcional, esta lista se usa en el caso de que la operación que se vaya a realizar sea un borrado o una modificación. En el caso de las modificaciones, contendrá el valor antiguo que se desea cambiar (el WHERE de la operación UPDATE), en el borrado también se trata de la cláusula WHERE de la query. Para hacer dicha operaciones se rellena la etiqueta ***campo_busqueda*** asignando en ***nombre*** el valor del campo de la tabla y su valor.
- Es importante especificar que en estos parámetros cuando lo que se quiere tratar es un campo string este debe introducirse entre comillas (‘), como se puede ver en los casos de prueba.

- **Casos de Prueba**

Caso 1 – Realizar una inserción en una tabla.

```
<?xml version="1.0" encoding="UTF-8"?>
<content xsi:noNamespaceSchemaLocation="GestionTablas.xsd"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
        <request>
                <datos_validacion>
                        <usuario_valid>dmadmin</usuario_valid>
                        <psw_validacion>dmadmin</psw_validacion>
                        <docbase>ICM</docbase>
                </datos_validacion>
                <tabla>dm_dbo. AUX_TIPO_EXP</tabla>
                <operacion>INSERT</operacion>
                <lista_Atributos>
                        <campo nombre="CODE" clave=”true”>0</campo>
                        <campo nombre="DESCRIPCION" clave=”false”>’descripción ejg’</campo>
                </lista_Atributos>
                <lista_Busqueda></lista_Busqueda>
        </request>
</content>
```

Caso 2 – Realizar un modificacion de un campo de la tabla.

```
<?xml version="1.0" encoding="UTF-8"?>
<content xsi:noNamespaceSchemaLocation="GestionTablas.xsd"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
        <request>
                <datos_validacion>
                        <usuario_valid>dmadmin</usuario_valid>
                        <psw_validacion>dmadmin</psw_validacion>
                        <docbase>ICM</docbase>
                </datos_validacion>
                <tabla>dm_dbo. AUX_TIPO_EXP</tabla>
                <operacion>UPDATE</operacion>
```

```
                    <lista_Atributos>
                        <campo nombre="DESCRIPCION" clave=”false”>’descripción modificada’ </campo>
                    </lista_Atributos>
                    <lista_Busqueda>
                    <campo_busqueda nombre="DESCRIPCION">’descripción’ </campo_busqueda>
            </lista_Busqueda>
              </request>
</content>
```

Caso 3 – Borrar un campo de la tabla.

```
<?xml version="1.0" encoding="UTF-8"?>
<content xsi:noNamespaceSchemaLocation="GestionTablas.xsd"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
        <request>
                <datos_validacion>
                        <usuario_valid>dmadmin</usuario_valid>
                        <psw_validacion>dmadmin</psw_validacion>
                        <docbase>ICM</docbase>
                </datos_validacion>
                <tabla>dm_dbo. AUX_TIPO_EXP</tabla>
                <operacion>DELETE</operacion>
                <lista_Atributos></lista_Atributos>
                <lista_Busqueda>
                <campo_busqueda nombre="DESCRIPCION">’descripción modificada’</campo_busqueda>
            </lista_Busqueda>
              </request>
</content>
```

5.1.6 USO DEL SERVICIO DE CHECK DOCUMENTO

Este Servicio permite bloquear, desbloquear y registrar los cambios en un documento por un usuario previamente bloqueado por el mismo. La forma de llamar a este Servicio es llamar al servicio **CheckDocumento** con alguna de las operaciones siguientes: **checkin** (registro de documento)**, checkout** (bloqueo de documento), **cancel checkout** (desbloqueo de documento)**,** pasándole como parámetro un xml**.**

5.1.6.1 DESCRIPCION DEL METODO CHECKIN

Permite registrar los cambios que ha realizado un usuario en un documento, previamente bloqueado por el mismo

Object[] checkin(String strXml, DataHandler dhFile, Logger log) throws WSException

Un ejemplo de llamada al método ***checkin*** se encuentra en el siguiente código java. En este se indica la ruta temporal donde se encuentra el fichero xml, este fichero se pasa a un String que será un parámetro de entrada del método.

Ejemplo de llamada al método checkin

El fichero xml (toda su ruta) es previamente transformado en string como:

```
String ruta = Config.get(“aplicacion.download”)+" check.xml";
FileInputStream fileinputstream = new FileInputStream(fichruta);
byte abyte0[] = new byte[fileinputstream.available()];
fileinputstream.read(abyte0);
fileinputstream.close();
s = new String(abyte0);
s = s.trim();
//Crear el DataHandler a partir del documento físico
```

```
String fichero = Config.get("aplicacion.download")+" a.tif";
File fi = new File (fichero);
if (fi.exists()){
   dh = new DataHandler (new FileDataSource (fi.getAbsolutePath()));
}

Object[] obj = checkdocumento.checkin(s, dh, logger);
```

- **Parámetros de entrada:**

Parámetro	Descripción	Tipo	Obligatorio
strXml	Xml de entrada, según schema **CheckDocumento.xsd**	String	SI
dhFile	Objeto DataHandler con el contenido del documento modificado que se quiere registrar. Este Datahandler lo crea el cliente a partir del path del documento que se desea añadir como contenido.	DataHandler	NO
log	Objeto Logger para escribir los errores	Logger	Si

- **Retorno:**

Object[ ] – objeto, en el que en su primera posición se encuentra el DataHandler del contenido del documento si en los parámetros de entrada se especifica que se quiere guardar copia del mismo, la segunda posición del objeto estará informada con un XML que contiene la respuesta de la operación, indicando si la operacion ha sido correcta o no. Contiene tres parámetros: nombre_documento, resultado y mensaje. Si el valor del parámetro resultado es 0, indica que no ha habido ningún problema, en caso de tener otro valor, indica que se ha producido un error, descrito en el valor del parámetro mensaje. El parámetro *nombre_documento* contiene el nombre del documento Este xml sigue el schema **CheckDocumento.xsd.**

XML DE RETORNO

```
<?xml version="1.0" encoding="UTF-8"?>
<response>
    <resultado>0</resultado>
    <mensaje>String</mensaje>
    <nombre_documento>String</nombre_documento>
</response>
```

- **Schema**

Schema CheckDocumento.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XMLSpy v2007 (http://www.altova.com) by UF787648 (EMBRACE) -->
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
attributeFormDefault="unqualified">
```

```
    <xs:element name="request">
        <xs:complexType>
            <xs:all>
                <xs:element ref="datos_validacion"/>
                <xs:element name="id_documento" type="xs:string"/>
                <xs:element name="extension" type="xs:string" minOccurs="0"/>
                <xs:element name="nombreDocumento" type="xs:string" minOccurs="0"/>
                <xs:element name="descripcion" type="xs:string" minOccurs="0"/>
                <xs:element name="locked" type="xs:string" minOccurs="0"/>
                <xs:element name="version" type="xs:string" minOccurs="0"/>
                <xs:element name="labelVersion" type="xs:string" minOccurs="0"/>
                <xs:element name="currentVersion" type="xs:string" minOccurs="0"/>
                <xs:element name="subscribe" type="xs:string" minOccurs="0"/>
                <xs:element name="indexar" type="xs:string" minOccurs="0"/>
            </xs:all>
            <xs:attribute name="session" type="xs:string"/>
            <xs:attribute name="cerrar_sesion" type="xs:boolean"/>
        </xs:complexType>
    </xs:element>
    <xs:element name="datos_validacion">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="usuario_valid" type="xs:string"/>
                <xs:element name="psw_validacion" type="xs:string"/>
                <xs:element name="docbase" type="xs:string"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="response">
        <xs:complexType>
            <xs:all>
                <xs:element name="resultado" type="xs:string"/>
                <xs:element name="mensaje" type="xs:string" minOccurs="0"/>
                <xs:element name="nombre_documento" type="xs:string" minOccurs="0"/>
            </xs:all>
            <xs:attribute name="cerrar_sesion" type="xs:boolean"/>
            <xs:attribute name="session" type="xs:string"/>
        </xs:complexType>
    </xs:element>
    <xs:element name="content">
        <xs:complexType>
            <xs:choice>
                <xs:element ref="request"/>
                <xs:element ref="response"/>
            </xs:choice>
        </xs:complexType>
    </xs:element>
</xs:schema>
```

Indicaciones:

- Los parámetros ***<usuario_valid>, <psw_validacion> y <docbase>*** son necesarios para realizar las validaciones de usuario e iniciar la sesión de documentum,
- El parámetro **<id_documento>** es obligatorio
- Los siguientes parámetros son opcionales:
  - **<extension>,** parámetro para informar la extensión del nuevo contenido que se quiera dar al documento al hacer su registro
  - **<nombreDocumento>** indica el nombre con el cual el documento es guardado en la base de datos de Documentum, si este parámetro no está informado se quedará con el nombre anterior.
  - **<descripción>**, hace referencia a la descripción con la que se registra el documento.

- El parámetro **<locked>** si está informado con 'true' significa que el nuevo documento registrado debe permanecer bloqueado por el usuario.
- Mediante el parámetro **<version>** se indica la nueva versión con la que será registrado el documento, aquí se indicara, 'minor', 'same' (misma version), o 'major' , si esta informado **<labelVersion>** la versión registrado contemplará esta etiqueta, y si **<currentVersion>** está informado a 'true' la etiqueta CURRENT estará en el nuevo documento registrado .
- **<susbcribe>** e **<indexar>** son parámetro informados a 'true' o 'false' y significan si están informados a 'true' que se subscribirá el documento y que se indexará el texto del documento, respectivamente.

- **Casos de Prueba**

Caso 1: Registrar un documento especificando la ruta. Sin guardar copia

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<content>
  <request>
    <datos_validacion>
        <usuario_valid>dmadmin</usuario_valid>
        <psw_validacion>dmadmin</psw_validacion>
        <docbase>ICM</docbase>
    </datos_validacion>
    <id_documento>090017728000e2e7</id_documento>
    <extension>txt</extension>
    <nombreDocumento>Documento</nombreDocumento>
    <descripcion>Prueba de Checkin</descripcion>
    <locked>false</locked>
    <version>minor</version>
    <labelVersion>Version</labelVersion>
    <currentVersion>true</currentVersion>
    <subscribe>false</subscribe>
    <indexar>false</indexar>
  </request>
</content>
```

Caso 2: Registrar un documento, guardando copia y bloqueandolo.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<content>
  <request>
    <datos_validacion>
        <usuario_valid>dmadmin</usuario_valid>
        <psw_validacion>dmadmin</psw_validacion>
        <docbase>ICM</docbase>
    </datos_validacion>
    <id_documento>090017728000e28b</id_documento>
    <extension>pdf</extension>
    <nombreDocumentoDocumento Checkin</nombreDocumento>
    <descripcion>Prueba checkin dos</descripcion>
    <locked>true</locked>
    <version>major</version>
    <labelVersion>MAG</labelVersion>
    <currentVersion>true</currentVersion>
    <subscribe>false</subscribe>
    <indexar>false</indexar>
  </request>
</content>
```

5.1.6.2 DESCRIPCION DEL METODO CHECKOUT

Permite bloquear un documento.

Object[] checkout(String strXml, Logger log) throws WSException

Un ejemplo de llamada al método ***checkout*** se encuentra en el siguiente código java. En este se indica la ruta temporal donde se encuentra el fichero xml, este fichero se pasa a un String que será un parámetro de entrada del método.

Ejemplo de llamada al método checkout

El fichero xml (toda su ruta) es previamente transformado en string como:

```
String ruta = Config.get("aplicacion.download")+" check.xml";
FileInputStream fileinputstream = new FileInputStream(fichruta);
byte abyte0[] = new byte[fileinputstream.available()];
fileinputstream.read(abyte0);
fileinputstream.close();
s = new String(abyte0);
s = s.trim();

Object[] obj = checkDocumento.checkout(s, logger);
```

- **Parámetros de entrada:**

Parámetro	Descripción	Tipo	Obligatorio
strXml	Xml de entrada, según schema **CheckDocumento.xsd**	String	SI
log	Objeto Logger para escribir los errores	Logger	SI

- **Retorno:**

Object[ ] – objeto, en el que en su primera posición se encuentra el DataHandler del contenido del documento, que el cliente se encargará de transformar en un fichero físico para guardar en local, esta copia en local se hará en un sitio especifico que indique el cliente, por ejemplo :
dirDestino=**new** File(System.*getProperty*("user.home") + System.*getProperty*("file.separator") + "Checkout");
La segunda posición del objeto estará informada con un XML que contiene la respuesta de la operación, indicando si la operacion ha sido correcta o no. Contiene tres parámetros: nombre_documento, resultado y mensaje. Si el valor del parámetro resultado es 0, indica que no ha habido ningún problema, en caso de tener otro valor, indica que se ha producido un error, descrito en el valor del parámetro mensaje. El parámetro **<nombre_documento>** contiene el nombre del documento. Este xml sigue el schema **CheckDocumento.xsd**

XML DE RETORNO

```
<?xml version="1.0" encoding="UTF-8"?>
<response>
    <resultado>0</resultado>
    <mensaje>String</mensaje>
    <nombre_documento>String</nombre_documento>
</response>
```

El parámetro **<nombre_documento>** contiene el nombre del documento

- **Schema**

Schema CheckDocumento.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XMLSpy v2007 (http://www.altova.com) by UF787648 (EMBRACE) -->
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
attributeFormDefault="unqualified">
    <xs:element name="request">
        <xs:complexType>
            <xs:all>
                <xs:element ref="datos_validacion"/>
                <xs:element name="id_documento" type="xs:string"/>
                <xs:element name="extension" type="xs:string" minOccurs="0"/>
                <xs:element name="nombreDocumento" type="xs:string" minOccurs="0"/>
                <xs:element name="descripcion" type="xs:string" minOccurs="0"/>
                <xs:element name="locked" type="xs:string" minOccurs="0"/>
                <xs:element name="version" type="xs:string" minOccurs="0"/>
                <xs:element name="labelVersion" type="xs:string" minOccurs="0"/>
                <xs:element name="currentVersion" type="xs:string" minOccurs="0"/>
                <xs:element name="subscribe" type="xs:string" minOccurs="0"/>
                <xs:element name="indexar" type="xs:string" minOccurs="0"/>
            </xs:all>
            <xs:attribute name="session" type="xs:string"/>
            <xs:attribute name="cerrar_sesion" type="xs:boolean"/>
        </xs:complexType>
    </xs:element>
    <xs:element name="datos_validacion">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="usuario_valid" type="xs:string"/>
                <xs:element name="psw_validacion" type="xs:string"/>
                <xs:element name="docbase" type="xs:string"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="response">
        <xs:complexType>
            <xs:all>
                <xs:element name="resultado" type="xs:string"/>
                <xs:element name="mensaje" type="xs:string" minOccurs="0"/>
                <xs:element name="nombre_documento" type="xs:string" minOccurs="0"/>
            </xs:all>
            <xs:attribute name="cerrar_sesion" type="xs:boolean"/>
            <xs:attribute name="session" type="xs:string"/>
        </xs:complexType>
    </xs:element>
    <xs:element name="content">
        <xs:complexType>
            <xs:choice>
                <xs:element ref="request"/>
                <xs:element ref="response"/>
            </xs:choice>
        </xs:complexType>
    </xs:element>
</xs:schema>
```

Indicaciones:

- Los parámetros ***<usuario_valid>, <psw_validacion> y <docbase>*** son necesarios para realizar las validaciones de usuario e iniciar la sesión de documentum,

- El parámetro **<id_documento>** es obligatorio

- **Casos de Prueba**

Caso 1: Checkout pasándole identificador del documento

```
<?xml version="1.0" encoding="UTF-8"?>
<content xsi:noNamespaceSchemaLocation="checkout.xsd" xmlns:xsi="http://ww
w.w3.org/2001/XMLSchema-instance">
    <request>
        <datos_validacion>
            <usuario_valid>dmadmin</usuario_valid>
            <psw_validacion>dmadmin</psw_validacion>
            <docbase>ICM</docbase>
        </datos_validacion>
        <id_documento>0900177280002e7e</id_documento>
    </request>
</content>
```

5.1.6.3 DESCRIPCION DEL METODO CANCEL CKECKOUT

Permite desbloquear un documento previamente bloqueado con la operación Checkout. La forma de llamar a este Servicio es llamar al método **cancelCheckout** de la clase **checkdocumento** pasándole como parámetros, un string con el xml que contiene los datos necesarios para la ejecución y un objeto logger para escribir los errores.

String cancelCheckout(String strXml, Logger log)

Un ejemplo de llamada al método ***cancelCheckout*** se encuentra en el siguiente código java. En este se indica la ruta temporal donde se encuentra el fichero xml, este fichero se pasa a un String que será un parámetro de entrada del método.

Ejemplo de llamada al método cancelCheckout

El fichero xml (toda su ruta) es previamente transformado en string como:

```
String ruta = Config.get("aplicacion.download")+" check.xml";
FileInputStream fileinputstream = new FileInputStream(fichruta);
byte abyte0[] = new byte[fileinputstream.available()];
fileinputstream.read(abyte0);
fileinputstream.close();
s = new String(abyte0);
s = s.trim();

Object[] obj = checkDocumento.cancelCheckout(s, logger);
```

- **Parámetros de entrada:**

Parámetro	Descripción	Tipo	Obligatorio
strXml	Xml de entrada, según schema **CheckDocumento.xsd**	String	SI
log	Objeto Logger para escribir los errores	Logger	SI

- **Retorno:**

String - XML con la respuesta de la operación, indicando si la operacion ha sido correcta o no. Contiene tres parámetros: nombre_documento, resultado y mensaje. Si el valor del parámetro resultado es 0, indica que no ha habido ningún problema, en caso de tener otro valor, indica que se ha producido un error, descrito en el valor del parámetro mensaje. El parámetro nombre_documento contiene el nombre del documento. Este xml sigue el schema **CheckDocumento.xsd**

XML DE RETORNO

```
<?xml version="1.0" encoding="UTF-8"?>
<response>
    <resultado>0</resultado>
    <mensaje>String</mensaje>
    <nombre_documento>String</nombre_documento>
</response>
```

El cliente que llama a esta operación debe borrar la copia de local que ha dejado el cliente que llamó a la operación de Checkout, siempre y cuando ambas estén especificando la misma ruta:

```
dirDestino=new File(System.getProperty("user.home") + System.getProperty("file.separator") + "Checkout" );
```

- **Schema**

Schema CheckDocumento.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XMLSpy v2007 (http://www.altova.com) by UF787648 (EMBRACE) -->
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
attributeFormDefault="unqualified">
    <xs:element name="request">
        <xs:complexType>
            <xs:all>
                <xs:element ref="datos_validacion"/>
                <xs:element name="id_documento" type="xs:string"/>
                <xs:element name="extension" type="xs:string" minOccurs="0"/>
                <xs:element name="nombreDocumento" type="xs:string" minOccurs="0"/>
                <xs:element name="descripcion" type="xs:string" minOccurs="0"/>
                <xs:element name="locked" type="xs:string" minOccurs="0"/>
                <xs:element name="version" type="xs:string" minOccurs="0"/>
                <xs:element name="labelVersion" type="xs:string" minOccurs="0"/>
                <xs:element name="currentVersion" type="xs:string" minOccurs="0"/>
                <xs:element name="subscribe" type="xs:string" minOccurs="0"/>
                <xs:element name="indexar" type="xs:string" minOccurs="0"/>
            </xs:all>
            <xs:attribute name="session" type="xs:string"/>
            <xs:attribute name="cerrar_sesion" type="xs:boolean"/>
        </xs:complexType>
    </xs:element>
    <xs:element name="datos_validacion">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="usuario_valid" type="xs:string"/>
                <xs:element name="psw_validacion" type="xs:string"/>
                <xs:element name="docbase" type="xs:string"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
```

```
<xs:element name="response">
        <xs:complexType>
            <xs:all>
                <xs:element name="resultado" type="xs:string"/>
                <xs:element name="mensaje" type="xs:string" minOccurs="0"/>
                <xs:element name="nombre_documento" type="xs:string" minOccurs="0"/>
            </xs:all>
            <xs:attribute name="cerrar_sesion" type="xs:boolean"/>
            <xs:attribute name="session" type="xs:string"/>
        </xs:complexType>
</xs:element>
<xs:element name="content">
        <xs:complexType>
            <xs:choice>
                <xs:element ref="request"/>
                <xs:element ref="response"/>
            </xs:choice>
        </xs:complexType>
    </xs:element>
</xs:schema>
```

Indicaciones:

- Los parámetros ***<usuario_valid>, <psw_validacion> y <docbase>*** son necesarios para realizar las validaciones de usuario e iniciar la sesión de documentum,
- El parámetro **<id_documento>** es obligatorio, se corresponde con el identificador del documento que se desea desbloquear

- **Casos de Prueba**

Caso 1: Cancelar Checkout pasándole identificador del documento

```
<?xml version="1.0" encoding="UTF-8"?>
<content xsi:noNamespaceSchemaLocation="cancelarcheckout.xsd" xmlns:xsi="http://ww
w.w3.org/2001/XMLSchema-instance">
        <request>
            <datos_validacion>
                <usuario_valid>dmadmin</usuario_valid>
                <psw_validacion>dmadmin</psw_validacion>
                <docbase>ICM</docbase>
            </datos_validacion>
            <id_documento>09001772800028fe</id_documento>
        </request>
</content>
```

5.1.7 USO DEL SERVICIO DE IMPORTAR DOCUMENTO

Este Service permite importar y modificar documentos. Permite modificar tanto metadatos como contenido. La forma de llamar a este Servicio es llamar al método **ImportarModificarDoc** de la clase **importarmodificar** pasándole como parámetros, un string con el xml que contiene los datos necesarios para la ejecución (metadatos del documento), un objeto datahandler con el contenido del documento y un objeto logger para escribir los errores.

5.1.7.1 DESCRIPCION DEL METODO

Permite importar y modificar documentos

String ImportarModificarDoc(String strXml, DataHandler dhFile, Logger log) throws WSException.

Un ejemplo de llamada al método **ImportarModificarDoc** se encuentra en el siguiente código java. En este se indica la ruta temporal donde se encuentra el fichero xml, este fichero se pasa a un String que será un parámetro de entrada del método.

Ejemplo de llamada al método ImportarModificarDoc

El fichero xml (toda su ruta) es previamente transformado en string como:

```
String ruta = Config.get("aplicacion.download")+" ImportarModificar.xml";
FileInputStream fileinputstream = new FileInputStream(fichruta);
byte abyte0[] = new byte[fileinputstream.available()];
fileinputstream.read(abyte0);
fileinputstream.close();
s = new String(abyte0);
s = s.trim();

//Generacion del Datahandler del fichero:
String fichero =  Config.get("aplicacion.download")+" peque.txt";
File fi = new File (fichero);
if (fi.exists()){
   dh = new DataHandler (new FileDataSource (fi.getAbsolutePath()));
}
Object[] obj = importarmodificar.ImportarModificarDoc(s, dh, logger);
```

- **Parámetros de entrada:**

Parámetro	Descripción	Tipo	Obligatorio
strXml	Xml de entrada, según schema **ImportarModificar.xsd**	String	SI
dhFile	Objeto DataHandler con el contenido del documento a importar o modificar. Este Datahandler lo crea el cliente a partir del path del documento que se desea añadir como contenido.	DataHandler	NO
log	Objeto Logger para escribir los errores	Logger	SI

- **Retorno:**

String - XML con la respuesta de la operación, indicando si la operacion ha sido correcta o no. Contiene dos parámetros: resultado y mensaje. Si el valor del parámetro resultado es 0, indica que no ha habido ningún problema, en caso de tener otro valor, indica que se ha producido un error, descrito en el valor del parámetro mensaje. En el parámetro idDocumento devuelve el id del documento importado. Este xml sigue el schema **ImportarModificar.xsd**

XML DE RETORNO

```
<?xml version="1.0" encoding="UTF-8"?>
<content>
    <response>
        <idDocumento>String</idDocumento>
        <resultado>0</resultado>
        <mensaje>String</mensaje>
    </response>
</content>
```

- **Schema**

Schema ImportarModificar.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XMLSpy v2007 (http://www.altova.com) by UF787648 (EMBRACE) -->
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
attributeFormDefault="unqualified">
    <xs:element name="response">
        <xs:complexType>
            <xs:all>
                <xs:element ref="mensaje"/>
                <xs:element ref="resultado"/>
                <xs:element ref="idDocumento" minOccurs="0"/>
            </xs:all>
            <xs:attribute name="sesion" type="xs:string"/>
<xs:attribute name="cerrarSesion" type="xs:boolean"/>
        </xs:complexType>
    </xs:element>
    <xs:element name="docBase" type="xs:string"/>
    <xs:element name="pswValid" type="xs:string"/>
    <xs:element name="usuarioValid" type="xs:string"/>
    <xs:element name="datosValidacion">
        <xs:complexType>
            <xs:all>
                <xs:element ref="usuarioValid"/>
                <xs:element ref="pswValid"/>
                <xs:element ref="docBase"/>
            </xs:all>
        </xs:complexType>
    </xs:element>
    <xs:element name="atributo">
        <xs:complexType>
            <xs:simpleContent>
                <xs:extension base="xs:string">
                    <xs:attribute name="nombre" type="xs:string"/>
                    <xs:attribute name="valorRepeating" type="xs:string"/>
                    <xs:attribute name="operacion" type="xs:string"/>
                    <xs:attribute name="indiceRepeating" type="xs:string"/>
                </xs:extension>
            </xs:simpleContent>
        </xs:complexType>
    </xs:element>
   <xs:element name="extension" type="xs:string"/>
    <xs:element name="listaAtributos">
        <xs:complexType>
            <xs:sequence>
                <xs:element ref="atributo" maxOccurs="unbounded"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="mensaje" type="xs:string"/>
    <xs:element name="resultado" type="xs:int"/>
    <xs:element name="idDocumento" type="xs:string"/>
    <xs:element name="request">
        <xs:complexType>
            <xs:all>
                <xs:element ref="datosValidacion"/>
                <xs:element ref="extension" minOccurs="0"/>
```

```
                <xs:element ref="tipoDocumental"/>
                <xs:element ref="rutaRepositorio" minOccurs="0"/>
                <xs:element ref="listaAtributos"/>
                <xs:element ref="acl" minOccurs="0"/>
                <xs:element ref="idDocumento" minOccurs="0"/>
            </xs:all>
            <xs:attribute name="sesion" type="xs:string"/>
            <xs:attribute name="cerrarSesion" type="xs:boolean"/>
        </xs:complexType>
    </xs:element>
    <xs:element name="tipoDocumental" type="xs:string"/>
    <xs:element name="rutaRepositorio" type="xs:string"/>
    <xs:element name="acl">
        <xs:complexType>
            <xs:simpleContent>
                <xs:extension base="xs:string">
                    <xs:attribute name="nombre" type="xs:string"/>
                    <xs:attribute name="dominio" type="xs:string"/>
                </xs:extension>
    </xs:simpleContent>
        </xs:complexType>
    </xs:element>
    <xs:element name="content">
        <xs:annotation>
            <xs:documentation>Comment describing your root element</xs:documentation>
        </xs:annotation>
        <xs:complexType>
            <xs:choice>
                <xs:element ref="request"/>
                <xs:element ref="response"/>
            </xs:choice>
        </xs:complexType>
    </xs:element>
</xs:schema>
```

Indicaciones:

- Los parámetros ***<usuario_valid>, <psw_validacion> y <docbase>*** son necesarios para realizar las validaciones de usuario e iniciar la sesión de documentum,
- El parámetro **<tipo documental>**, es obligatorio. Para indicar el tipo documental en el que va a tener lugar la importación o la modificación.
- El parámetro **<listaAtributos>**, es obligatorio. En este parámetro se irán informando de los distintos atributos, el campo **operacion** sólo permite los valores **ADD, MODIFY o REMOVE**, los campos **valorRepeating** e **indiceRepeating** son excluyentes, el servicio comprueba siempre el segundo y en caso de estar informado no evalúa el primero. El web service comprueba los distintos atributos de cada tipo documental en el tipo documental, dm_type.
- Los siguientes parámetros son opcionales:
  - **<idDocumento>**, estará informado solo si se trata de una modificación.
  - **<acl>** parámetro en el que se especifican los permisos del documento, si no se especifica nada, por defecto el documento tendrá los permisos de la carpeta.
  - **<ruta_repositorio>** para indicar la ruta de carpetas donde se importará el documento o donde se haya el documento a modificar.
  - **<extensión>** para indicar la extensión del documento a importar. El documento a importar se informa como un Datahandler al servicio, dicho Datahandler será creado por el cliente.

- **Casos de Prueba**

Caso 1: Modificación de los permisos de un documento

```
<?xml version="1.0" encoding="UTF-8"?>
        <content xsi:noNamespaceSchemaLocation="permisosdocumento.xsd"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
                <request>
                        <datos_validacion>
                                <usuario_valid>dmadmin</usuario_valid>
                                <psw_validacion>dmadmin</psw_validacion>
                                <docbase>ICM</docbase>
                        </datos_validacion>
                        <id_documento>090017728000e538</id_documento>
                        <permisos>
                          <usuario_documento permiso="Delete" nombre="dmadmin" permiso_extendido =
        “DELETE_OBJECT”/>
                        </permisos>
                         <restricciones>
                         </restricciones>

                </request>
        </content>
```

5.1.8 USO DEL SERVICIO DE GESTION DE CARPETAS

Este Servicio permite añadir, modificar y borrar cabinet y fólder. La forma de llamar a este Servicio es llamar al método **gestoncarpetas** de la clase **gestioncarpetas** pasándole como parámetros, un string con el xml que contiene los datos necesarios para la ejecución y un objeto logger para escribir los errores.

5.1.8.1 DESCRIPCION DEL METODO

Permite añadir, modificar y borrar cabinet y fólder

String gestionCarpetas(String strXml, Logger log)

Un ejemplo de llamada al método **gestionCarpetas** se encuentra en el siguiente código java. En este se indica la ruta temporal donde se encuentra el fichero xml, este fichero se pasa a un String que será un parámetro de entrada del método.

Ejemplo de llamada al método gestionCarpetas

El fichero xml (toda su ruta) es previamente transformado en string como:

```
String ruta = "/opt/app/bea/user_projects/domains/icmws/myserver/stage/_appsdir_icm2_war/icm2.war/tmp_dir/gestionTablas.xml";
FileInputStream fileinputstream = new FileInputStream(fichruta);
byte abyte0[] = new byte[fileinputstream.available()];
fileinputstream.read(abyte0);
fileinputstream.close();
s = new String(abyte0);
s = s.trim();

Object[] obj = gestiontablas.gestiontablas(str_conexion, s, logger);
```

- **Parámetros de entrada:**

Parámetro	Descripción	Tipo	Obligatorio
strXml	Xml de entrada, según schema **GestionCarpetas.xsd**	String	SI
log	Objeto Logger para escribir los errores	Logger	SI

- **Retorno:**

String - XML con la respuesta de la operación, indicando si la operacion ha sido correcta o no. Contiene dos parámetros: resultado y mensaje. Si el valor del parámetro resultado es 0, indica que no ha habido ningún problema, en caso de tener otro valor, indica que se ha producido un error, descrito en el valor del parámetro mensaje. Este xml sigue el schema **GestionCarpetas.xsd**

XML DE RETORNO

```
<?xml version="1.0" encoding="UTF-8"?>
<content>
    <response>
        <resultado>0</resultado>
        <mensaje>String</mensaje>
    </response>
</content>
```

- **Schema**

Schema GestionCarpetas.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XMLSPY v2004 rel. 3 U (http://www.xmlspy.com) by Fernando Humanes Pereira
(Comunidad de Madrid) -->
<!--Sample XML file generated by XMLSPY v2004 rel. 3 U (http://www.xmlspy.com)-->
<content xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="D:\ENTREGAS\INDRA\DOCU_20080922\docu_util_lib\fuentes\xsd\
GestionCarpetas.xsd">
    <request sesion="String" cerrarSesion="1">
        <datosValidacion>
            <usuarioValid>String</usuarioValid>
            <pswValid>String</pswValid>
            <docBase>String</docBase>
        </datosValidacion>
        <idObj>String</idObj>
        <tipoObj>String</tipoObj>
        <nombreObj>String</nombreObj>
        <pathPadre>String</pathPadre>
        <aclDomain>String</aclDomain>
        <aclName>String</aclName>
        <operacion>String</operacion>
```

```
        <atributos>
          <atributo nombre="String" operacion="String" valorRepeating="String"
indiceRepeating="String">String</atributo>
        </atributos>

        <borrarContenido>String</borrarContenido>
        <pathOriginal>String</pathOriginal>
    </request>
</content>
```

Indicaciones:

- Los parámetros ***<usuario_valid>, <psw_validacion> y <docbase>*** son necesarios para realizar las validaciones de usuario e iniciar la sesión de documentum,
- El parámetro ***<operacion>*** es obligatorio, a través del cual indicamos la operación que queremos realizar, pudiendo tomar los valores de Alta, Baja o Modificación es obligatorio
- Los siguientes parámetros son opcionales:
  - ***<idObj>***, para indicar el identificador del documento sobre el cual se realizará una operación de baja o modificación.
  - ***<tipoObj>***, para indicar si se creará un cabinet o una fólder. Es importante recordar que los cabinet (dm_cabinet) extienden de las folders (dm_folder).
  - ***<nombreObj>***, indica el nombre del objeto que se desea modificar.
  - ***<pathPadre>***, se utiliza en el alta para indicar donde situar la carpeta. En el caso de las cabinet no es necesario.
  - ***<aclDomain>***, para indicar el dominio de la ACL, que se asignara al objeto a crear o modificar.
  - ***<aclName>***, para indicar el ACL que se asignará al objeto a crear o modificar.
  - ***<borrarContenido>***, se utiliza para forzar el borrado de una fólder o cabinet independientemente de si tiene o no contenido.
  - ***< pathOriginal>***, para indicar el path original de una carpeta en las operaciones en las que quieres cambiar una carpeta de una ubicación a otra.
  - Lista ***<atributos>***, para modificar los valores de los campos que se desee en las tablas dm_cabinet o dm_folder. Para ello solo se escribe en la etiqueta ***atributo***, en el campo ***nombre***, el nombre del campo de la tabla y asignarle el valor deseado, el campo ***operación*** sólo permite los valores **ADD, MODIFY o REMOVE**, los campos ***valorRepeating*** e ***indiceRepeating*** son excluyentes, el servicio comprueba siempre el segundo y en caso de estar informado no evalúa el primero.

- **Casos de Prueba**

Caso 1: Crear un cabinet.

```
<?xml version="1.0" encoding="UTF-8"?>
<content xsi:noNamespaceSchemaLocation="GestionCarpetas.xsd"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
        <request>
                <datosValidacion>
                        <usuarioValid>dmadmin</usuarioValid>
                        <pswValid>dmadmin</pswValid>
                        <docBase>ICM</docBase>
                </datosValidacion>
```

```
                <tipoObj>dm_cabinet</tipoObj>
                <nombreObj>CABINETejg</nombreObj>
                <aclDomain>dm_dbo</aclDomain>
                <aclName>dm_acl_superusers</aclName>
                <operacion>alta</operacion>
                <atributos>
                        <atributo nombre="subject">ejg</atributo>
                        <atributo nombre="title">cabinet uf</atributo>
                </atributos>
        </request>
</content>
```

Caso 2: Modificar una fólder

```
<?xml version="1.0" encoding="UTF-8"?>
<content xsi:noNamespaceSchemaLocation="GestionCarpetas.xsd"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
        <request>
                <datosValidacion>
                        <usuarioValid>dmadmin</usuarioValid>
                        <pswValid>dmadmin</pswValid>
                        <docBase>ICM</docBase>
                </datosValidacion>
                <idObj>0b0017728000ed49</idObj>
                <tipoObj>dm_folder</tipoObj>
                <nombreObj>Folder300407_2_modif</nombreObj>
                <pathPadre>/CABINETejg</pathPadre>
                <aclName>Global User Default ACL</aclName>
                <operacion>modificacion</operacion>
                <atributos>
                        <atributo nombre="subject">ejg 2_modificado</atributo>
                        <atributo nombre="title">titulo_modificado</atributo>
<atributo nombre="atr_repeating" operación=”MODIFY” valorRepeating=”valor1” >valorNuevo
</atributo>
                </atributos>
        </request>
</content>
```

Caso 3: Eliminar un cabinet.

```
<?xml version="1.0" encoding="UTF-8"?>
<content xsi:noNamespaceSchemaLocation="GestionCarpetas.xsd"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
        <request>
                <datosValidacion>
                        <usuarioValid>dmadmin</usuarioValid>
                        <pswValid>dmadmin</pswValid>
                        <docBase>ICM</docBase>
                </datosValidacion>
                <idObj>0c00177280004bc9</idObj>
                <operacion>baja</operacion>
        </request>
</content>
```

5.1.9 USO DEL SERVICIO DE VER RENDITION

Este Servicio permite consultar las renderizaciones de los documentos. La forma de llamar a este Servicio es llamar al método **verRendition** de la clase **VerRendition** pasándole como parámetros, un string con el xml que contiene los datos necesarios para la ejecución y un objeto logger para escribir los errores.

El retorno de este Service es un objeto en el cual en su primera posición está el Datahandler del contenido de la renderización del documento, que el cliente se encargará de transformar en un fichero físico para guardar en local y visualizarlo, y en la segunda posición devuelve un xml en el que nos indica el resultado de la operación realizada.

5.1.9.1 DESCRIPCION DEL METODO

Permite visualizar documentos

Object[] oRespuesta = verRendition.verRendition(String strXML, Logger log, beanResultadoBusqueda bean);

Un ejemplo de llamada al método ***verRendition*** se encuentra en el siguiente código java. En este se indica la ruta temporal donde se encuentra el fichero xml, este fichero se pasa a un String que será un parámetro de entrada del método.

Ejemplo de llamada al método verRendition

El fichero xml (toda su ruta) es previamente transformado en string como:

```
String ruta = Config.get("aplicacion.download")+" verRendition.xml";
FileInputStream fileinputstream = new FileInputStream(fichruta);
byte abyte0[] = new byte[fileinputstream.available()];
fileinputstream.read(abyte0);
fileinputstream.close();
s = new String(abyte0);
s = s.trim();

Object[] obj = ver.verDocumento(s, logger);
```

- **Parámetros de entrada:**

Parámetro	Descripción	Tipo	Obligatorio
strXml	Xml de entrada, según schema **VerRendition.xsd**	String	SI
log	Objeto Logger para escribir los errores	Logger	SI
bean	Objeto para almacenar el resultado de la búsqueda	beanResultadoBusqueda	SI

- **Retorno:**

Object[ ] – objeto, en el que en su primera posición se encuentra el DataHandler de la renderización del documento que se quiere visualizar y que el cliente transformará en un fichero físico que guardará en local y visualizará, la segunda posición del objeto estará informada con un XML que contiene la respuesta de la operación, indicando si la operacion ha sido correcta o no. Contiene tres parámetros: resultado, mensaje y

formato. Si el valor del parámetro resultado es 0, indica que no ha habido ningún problema, en caso de tener otro valor, indica que se ha producido un error, descrito en el valor del parámetro mensaje. El parámetro *formato* contiene el formato de la renderización consultada. Este xml sigue el schema **VerRendition.xsd**

XML DE RETORNO

```
<?xml version="1.0" encoding="UTF-8"?>
<content>
    <response>
        <resultado>0</resultado>
        <mensaje>String</mensaje>
        <formato>String</formato>
    </response>
</content>
```

- **Schema VerDocumento.xsd**

Schema VerRendition.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
attributeFormDefault="unqualified">
    <xs:element name="request">
        <xs:complexType>
            <xs:all>
                <xs:element ref="datos_validacion"/>
                <xs:element ref="id_rendicion" type="xs:string"/>
                </xs:all>
            <xs:attribute name="session" type="xs:string"/>
            <xs:attribute name="cerrar_sesion" type="xs:boolean"/>
        </xs:complexType>
    </xs:element>
    <xs:element name="datos_validacion">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="usuario_valid" type="xs:string"/>
                <xs:element name="psw_validacion" type="xs:string"/>
                <xs:element name="docbase" type="xs:string"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="response">
        <xs:complexType>
            <xs:all>
                <xs:element ref="resultado" type="xs:string"/>
                <xs:element ref="mensaje" type="xs:string"/>
                <xs:element ref="formato" type="xs:string" miniOccurs=”0” />
            </xs:all>
            <xs:attribute name="cerrar_sesion" type="xs:boolean"/>
            <xs:attribute name="session" type="xs:string"/>
        </xs:complexType>
```

```
</xs:element>
    <xs:element name="content">
        <xs:complexType>
            <xs:choice>
                <xs:element ref="request"/>
                <xs:element ref="response"/>
            </xs:choice>
        </xs:complexType>
    </xs:element>
    <xs:element name="id_rendicion" type="xs:string"/>
    <xs:element name="resultado" type="xs:string"/>
    <xs:element name="mensaje" type="xs:string"/>
    <xs:element name="formato" type="xs:string"/>
</xs:schema>
```

Indicaciones:

- Los parámetros ***<usuario_valid>, <psw_validacion> y <docbase>*** son necesarios para realizar las validaciones de usuario e iniciar la sesión de documentum,
- Los parámetros **<id_rendition>** y **<formato>** son obligatorios.

- **Casos de Prueba**

Caso 1 –Visualización de Rendición

```
<?xml version="1.0" encoding="UTF-8"?>
<content xsi:noNamespaceSchemaLocation="verRenditionsara.xsd" xmlns:xsi="http://ww
w.w3.org/2001/XMLSchema-instance">
    <request>
        <datos_validacion>
            <usuario_valid>dmadmin</usuario_valid>
            <psw_validacion>dmadmin</psw_validacion>
            <docbase>ICM</docbase>
        </datos_validacion>
        <id_rendicion>0900177280004ba2</id_rendicion>
    </request>
</content>
```

5.2 PAQUETES COMUNES

En este apartado se describen las clases java comunes utilizados por los distintos Servicios.

5.2.1 Paquete docu_lib.icm.config

Clase	Descripción
LectorFormatos.java	Para llevar a cabo la validación de los formatos de los documentos que se importan, modifican y visualizan.

EvitarSqlInyectado.java	Se lleva a cabo la validación de que no existan parámetros de los xml de entrada que puedan llevar sql embebido. Lo que se hace es que en dichos parámetros no existan palabras clave que están presentes en una Query, como son “select ”,” where ”,”delete”, “update”, etc...
DefinicionFormatos.xml	Ficherio xml que define los formatos permitidos para los documentos.

5.2.2 Paquete docu_lib.icm.util.constants

Clase	Descripción
ErrorMessageConstants.java	Clase que contiene las constantes que van a definir el código de error y el mensaje que se asigna a una excepción. Esta formado por constantes de tipo String [] en el que el valor de la posición 0 es el código que se le asigna a la excepción y en la posición 1 se encuentra la descripción que la queremos dar
XMLParametersNameContanst.java	Clase de constantes que contiene literales con los nombres de los parámetros u otros atributos que se utilizan para leer algunos xml.

5.2.3 Paquete docu_lib.icm.util.exception

Clase	Descripción
WSException.java	Clases que extienden de la clase Exception. Tienen dos atributos: *mensaje y código*

Métodos clase WSException

```
/*constructores de la clase WSException*/
public WSException(String strMsj,String strCod);
/*constructores de WSException que además guarda la excepción a partir de la que hemos lanzado la actual*/
public WSException(String strMsj,String strCod,Exception eException);
/*devuelve el código asignado a la excepción*/
public String getCodigo();
```

```
/*modifica el código asignado a la excepción*/
public void setCodigo(String strCodigo;

/*devuelve el mensaje asignado a la excepción*/
public String getMensaje();

/*modifica el mensaje asignado a la excepción*/
public void setMensaje(String strMensaje);
```

Todas las excepciones lanzadas por los servicios serán recogidas y lanzadas en forma de ***WSException*** si se producen en el servicio de búsqueda.

5.2.4 Paquete docu_lib.icm.xml

Conjunto de clases y paquetes autogenerados a partir de la herramienta JAXB, utilizados para el manejo del fichero **DefinicionFormatos.xml** situado en **docu_lib.icm.config**. Todas estas clases y paquetes son de uso interno de la librería.

6 SOLUCION DE INTEGRACION DIRECTA CON DFC'S

6.1 INSTALACION Y CONFIGURACION

6.1.1 LIBRERÍAS NECESARIAS

Los proyectos que utilizan las DFC's, deben incluir los .jar utilizados en el directorio **lib** de compilación y de ejecución de la aplicación.
La versión de DFC's que se debe de utilizar es la 6.0 SP1 para Linux
Las dfc's de Documentum 6.0 SP1 para Linux se han empaquetado en **DFC_6.0_jar_linux.rar** y se han publicado en:
https://gestiona.madrid.org/soja_int/run/j/EnlaceLibreria.icm?cd_elemento=722

6.1.2 INSTALACION

Para desarrollar aplicaciones j2ee con acceso a documentum 6.0 vía DFC'S no es necesario tener instaladas las dfc's en la máquina de ejecución de la aplicación. Los .jar utilizados deben meterse en el directorio **lib** de compilación y de ejecución de la aplicación.
La jdk que debe utilizarse para el desarrollo de estos aplicativos con Documentum 6.0 es la indicada por el producto (**JDK-1.5.0_12**) ó superior.

6.1.3 CONFIGURACIÓN

Las variables que pueden venir en el fichero de configuración de la aplicación son las indicadas en la siguiente tabla.

Variables	Descripción	Valores que toma	Obligatorio
documentum.docbase	Nombre del repositorio	desdocum_01	SI
documentum.usuario	Normalmente la autenticación en documentum se realizará a través de LDAP. Esta variable sólo vendrá en el caso de necesitar un usuario genérico de conexión con Documentum.	xxxx_<usuario> siendo xxxx el nombre del proyecto	NO
documentum.clave	Clave del usurio encriptada del usuario genérico.		NO

NOTA:
Todas las variables se declararán en el apartado
/*************************/
/* DOCUMENTUM */
/*************************/.

Cualquier otra variable que se necesite definir seguirá la siguiente nomenclatura **documentum.<descripción_variable>**

Para la correcta ejecución de los proyectos que utilicen esta solución, se debe incluir el fichero dfc.properties en el directorio classes de la aplicación que la utilice.
A continuación se muestra un ejemplo de fichero dfc.properties en el que se muestran las variables que como mínimo deben informarse.

dfc.properties

```
dfc.docbroker.host[0]=icmcs03
dfc.docbroker.port[0]=1489
dfc.globalregistry.repository=desdocum_01
dfc.globalregistry.username=dm_bof_registry
dfc.globalregistry.password=GrRNPhLJrkoTDAZE0RGJow\=\=
```

6.1.4 USO

Para el desarrollo de estos aplicativos deben seguirse las recomendaciones proporcionadas en el documento ***Documentum Foundation Classes V6. Development Guide***
En estos aplicativos toda sesión abierta con documentum dentro de una accion, debe liberarse antes de finalizar dicha acción. Para obtener sesiones con documentum se utilizará Session Manager. La obtención de la sesión y el código que realice las tareas en el repositorio estarán dentro de un ***try-catch*** y en el ***finally*** se incluirá una llamada al método ***release*** para liberar la sesión abierta con Documentum.
Cada sesión obtenida de documentum debe liberarse con el método ***release*** dentro de una cláusula ***finally***

Ejemplo

```
import com.documentum.com.DfClientX;
import com.documentum.fc.client.IDfClient;
import com.documentum.fc.client.IDfSessionManager;
import com.documentum.fc.common.DfException;

....
private IDfSessionManager mySessMgr = null;

public static void demoConexionDesconexion
   (String repository, String userName, String password) throws DfException
{
IDfSession mySession = null;

/*  Obtener una Sesión con un Identificador */
DfClientX clientx = new DfClientX();
IDfClient client = clientx.getLocalClient();
mySessMgr = client.newSessionManager();
IDfLoginInfo logininfo = clientx.getLoginInfo();
loginInfo.setUser(userName);
loginInfo.setPassword(password);
mySessMgr.setIdentity(repository, loginInfo);

IDfSession mySession = null;

try
{

/*  Obtener una Sesión utilizando un método de Session Manager */
mySession = mySessMgr.getSession(repository);

/*  Insertar código que realice tareas en el  repositorio */
}catch(Exception e){

/*  Tratamiento de excepciones  */

sistemas.util.Trazas.impLog("Error en demoConexionDesconexion.");
sistemas.util.Trazas.imprimeErrorExtendido(e);
throw new Exception(e.getMessage(),e.getCause());

finally
{
/* liberar la sesión  */
if (null != mySessMgr && null != mySession) {
        mySessMgr.release(mySession);
}
}
}
```

7 LIBRERÍA DOCU_UTIL_LIB. TRATAMIENTO DE FICHEROS XML

Para la creación y manejo de los ficheros xml que se le pasarán como parámetro a cada uno de los servicios web en el caso de utilizar la solución **docu_ws** ó a cada uno de los métodos en el caso de utilizar la solución **docu_lib** se proporciona la librería **docu_util_lib.jar** que mediante el uso de JAXB realiza las operaciones necesarias para el manejo de información mediante XML.
Esta librería se encuentra publicada en:
https://gestiona.madrid.org/soja_int/run/j/EnlaceRecurso.icm?cd_recurso=1182
Para generar estos XML se tienen que seguir las especificaciones propuestas por ICM y que se detallan en el **Manual_manejo_XML.pdf** publicado en:
https://gestiona.madrid.org/soja_int/run/j/EnlaceManual.icm?cd_manual=781
Estos xml además deben cumplir los esquemas de los mismos publicados como ***Esquemas del FrameWork de servicios Documentales*** en https://gestiona.madrid.org/soja_int/run/j/EnlaceRecurso.icm?cd_recurso=1142.
La librería docu_util_lib es una librería que debe estar en la parte servidora, donde se desplieguen los servicios y debe ser una librería de compilación y ejecución de cada proyecto que vaya a usar los servicios documentales ó bien la librería docu_lib
A continuación se explica con un ejemplo como generar el XML de ida, que alimenta a los servicios y como leer el XML que devuelven los servicios.

7.1 GENERACIÓN XML DE IDA

XML DE IDA

```
String strXml = "";
docu_util_lib.<Servicio>.ObjectFactory of = null;
of = new docu_util_lib. <Servicio>.ObjectFactory();
docu_util_lib. <Servicio>.Content servicioContent = null;
docu_util_lib. <Servicio>.Request servicioRequest = null;
docu_util_lib. <Servicio>.DatosValidacion servicioValidacion = null;

try {
   servicioContent = (docu_util_lib.<Servicio>.Content)    of.createContent();
   servicioRequest = (docu_util_lib.<Servicio>.Request) of.createRequest();
   servicioValidacion = (docu_util_lib.<Servicio>.DatosValidacion) of.createDatosValidacion();

   servicioContent.setRequest(servicioRequest);
   servicioRequest.setDatosValidacion(servicioValidacion);
   servicioRequest.setDql(strDql);
   servicioValidacion.setUsuarioValid(login);
   servicioValidacion.setPswValidacion(pass);
   servicioValidacion.setDocbase(docbase);

  JAXBContext jc = JAXBContext.newInstance(<nombre del servicio>);
  Marshaller marshall = jc.createMarshaller();
  marshall.setEventHandler(new ValidationEventCollector());
  ByteArrayOutputStream baos = null;
  baos = new ByteArrayOutputStream();
   marshall.setProperty(Marshaller.JAXB_NO_NAMESPACE_SCHEMA_LOCATION,
   <nombre del esquema>);
   marshall.setProperty(Marshaller.JAXB_FORMATTED_OUTPUT, Boolean.FALSE);
   marshall.marshal(content, baos);
   xmlReturn = baos.toString("UTF-8");
        .
        .
        }
```

7.2 MANEJO DEL XML DE VUELTA

XML DE VUELTA

```
ValidationEventCollector vec = new ValidationEventCollector();
JAXBContext jc;
docu_util_lib.BuscarDocumento.Content content = null;
docu_util_lib.BuscarDocumento.Response response = null;
InputStream is = null;
int result = 0;
try {
      jc = JAXBContext.newInstance(<nombre del servicio>);
      Unmarshaller u = jc.createUnmarshaller();
      u.setEventHandler(vec);
      is = new ByteArrayInputStream(xmlresultado.getBytes("UTF-8"));
      u.setValidating(true);
      content = (docu_util_lib.<Servicio>.Content) u.unmarshal(is);
      response = (docu_util_lib.<Servicio>.Response) content.getResponse();
      result = response.getResultado();

      if (result == 0) {
            List listaResultados = null;
            Iterator iterator = null;
            docu_util_lib.<Servicio>.Columna columna = null;
            docu_util_lib.<Servicio>.ResultadoConsulta
            resultadoConsulta = null;
            docu_util_lib.<Servicio>.ListaFila fila = null;
            resultadoConsulta = (docu_util_lib.<Servicio>.ResultadoConsulta);
            response.getResultadoConsulta();
            List listaFila = resultadoConsulta.getListaFila();
            Iterator itListaFila = listaFila.iterator();
      .
      .
      }
```

8 PREGUNTAS MÁS FRECUENTES

Pregunta: ¿Dónde se encuentran desplegados los Webservices del Framework documentum en el entorno de desarrollo?

Respuesta: Se encuentran desplegados en http://desarrollo.madrid.org/docu_ws/services

Pregunta: ¿Cómo se llama la máquina del Content Server en el entorno de desarrollo?

Respuesta: La máquina de Content Server en desarrollo es **icmcs03,** puerto **1489**

9 ENLACES RELACIONADOS

Producto	URL	Versión
SOJA (Internet)	**https://gestiona.madrid.org/soja_int** (Se requiere usuario y password)	
SOJA (Intranet)	**http://desarrollo.madrid.org/soja_int**	
SOPORTE EMC:	**http://powerlink.emc.com.**	